QLOGIC

Simplify

Fast Fabric Users Guide

Information furnished in this manual is believed to be accurate and reliable. However, QLogic Corporation assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties which may result from its use. QLogic Corporation reserves the right to change product specifications at any time without notice. Applications described in this document for any of these products are for illustrative purposes only. QLogic Corporation makes no representation nor warranty that such applications are suitable for the specified use without further testing or modification. QLogic Corporation assumes no responsibility for any errors that may appear in this document.

No part of this document may be copied nor reproduced by any means, nor translated nor transmitted to any magnetic medium without the express written consent of QLogic Corporation.

Linux is a registered trademark of Linus Torvalds.

Microsoft and Windows are registered trademarks and Windows Server is a trademark of Microsoft Corporation.

Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc.

SUSE is a registered trademark of Novell, Inc.

All other brand and product names are trademarks or registered trademarks of their respective owners.

Document Revision History
Rev. A, 01/08/08

© 2008 QLogic Corporation. All rights reserved worldwide.
First Published: March, 2007
Printed in U.S.A.

QLogic Corporation, 26650 Aliso Viejo Parkway, Aliso Viejo, CA 92656, (800) 662-4471 or (949) 389-6000

Section 1 Introduction

1.1 Intended Audience . . . 1-1
1.2 License Agreements . . . 1-1
1.3 Technical Support . . . 1-2
1.3.1 Availability . . . 1-2
1.3.2 Contact Information . . . 1-2

Section 2 Fast Fabric Overview

2.1 Feature Overview . . . 2-1
2.2 Fast Fabric Architecture . . . 2-2
2.2.1 How Fast Fabric Works . . . 2-3

Section 3 Getting Started

3.1 Design the Fabric . . . 3-1
3.2 Set Up the Fabric . . . 3-2
3.3 Using Fast Fabric . . . 3-4
3.4 Installing and Verifying Firmware on the SilverStorm IB Chassis . . . 3-6
3.5 Installing and Configuring the Subnet Manager . . . 3-9
3.6 Installing and Verifying Firmware on the IB Switches . . . 3-10
3.7 Installing InfiniBand on the Remaining Servers . . . 3-12
3.8 Verifying InfiniBand on the Remaining Servers . . . 3-16
3.9 Complete Installation of additional IB Management Nodes . . . 3-18
3.10 Configure and Initialize Health Check Tools . . . 3-19
3.11 Running HPL . . . 3-20
3.12 Upgrading IB software . . . 3-21

Section 4 Fast Fabric TUI Menu

4.1 Host Setup via Fast Fabric . . . 4-3
4.1.1 Edit Configuration and Select/Edit Hosts Files . . . 4-4
4.1.2 Verify Hosts via Ethernet ping . . . 4-4
4.1.3 Verify RSH/RCP Configured . . . 4-4
4.1.4 Setup Password-less SSH/SCP . . . 4-5
4.1.5 Copy /etc/hosts to all hosts . . . 4-5
4.1.6 Show uname -a for all hosts . . . 4-5
4.1.7 Install/Upgrade QuickSilver Software . . . 4-5
4.1.8 Configure IPoIB IP Address . . . 4-6
4.1.9 Build MPI Test Apps and Copy to Hosts . . . 4-6
4.1.10 Reboot Hosts . . . 4-6
4.1.11 Refresh SSH Known Hosts . . . 4-6
4.1.12 Rebuild MPI Library and Tools . . . 4-6

4.1.13 Run a command on all hosts 4-7
4.1.14 Copy a file to all hosts 4-7
4.1.15 View ibtest result files 4-7
4.2 Host Admin via Fast Fabric. 4-8
4.2.1 Edit Config and Select/Edit Hosts Files 4-8
4.2.2 Verify Hosts via Ethernet Ping 4-8
4.2.3 Summary of Fabric Components 4-9
4.2.4 Show Status of Host IB Ports 4-9
4.2.5 Verify Hosts see each other 4-9
4.2.6 Verify Hosts ping via IPoIB 4-9
4.2.7 Refresh SSH Known Hosts 4-9
4.2.8 Check MPI Performance 4-10
4.2.9 Generate all Hosts Problem Report Info 4-10
4.2.10 Run a command on all hosts 4-10
4.2.11 View ibtest result files 4-10
4.3 QLogic IB Chassis Admin via Fast Fabric 4-11
4.3.1 Edit the Configuration and Select/Edit Chassis Files 4-11
4.3.2 Verify Chassis via Ethernet Ping 4-11
4.3.3 Update Chassis Firmware 4-12
4.3.4 Show Status of Chassis IB Ports 4-12
4.3.5 Reboot Chassis 4-13
4.3.6 Generate all Chassis Problem Report Information 4-13
4.3.7 Run a command on all chassis 4-13
4.3.8 View ibtest results files 4-13
4.4 SilverStorm Externally Managed IB Switch Administration via Fast Fabric 4-14
4.4.1 Edit Config and Select/Edit Chassis Files 4-14
4.4.2 Verify Switch via Firmware Dump 4-14
4.4.3 Update Switch Firmware 4-14
4.4.4 Reboot Switch 4-15
4.4.5 View ibtest result files 4-15

Section 5 Detailed Descriptions of Command LineTools

5.1 Common Tool Options 5-1
5.1.1 -? 5-1
5.1.2 -p 5-1
5.1.3 -S 5-2
5.1.4 -C 5-2
5.1.5 -n or -I 5-2
5.1.6 Selection of Hosts 5-3
5.1.7 Selection of Chassis 5-4

5.1.8 Selection of Switches 5-7
5.1.9 Selection of local Ports (subnets) 5-9
5.2 Basic Setup and Administration Tools 5-11
5.2.1 pingall 5-11
5.2.2 check_rsh 5-12
5.2.3 setup_ssh 5-13
5.2.4 cmdall 5-15
5.2.5 captureall 5-17
5.3 File Management Tools 5-19
5.3.1 scpall 5-19
5.3.2 uploadall 5-21
5.3.3 downloadall 5-22
5.3.4 Simplified Editing of Node-Specific Files 5-24
5.3.5 Simplified Setup of Node-Generic Files 5-24
5.4 Fabric Analysis Tools 5-25
5.4.1 Fabric_info 5-25
5.4.2 showallports 5-26
5.4.3 iba_report 5-28
5.4.4 saquery 5-56
5.5 Advanced Initialization and Verification - ibtest 5-60
5.5.1 ibtest Host Operations 5-63
5.5.2 ibtest Chassis Operations 5-66
5.5.3 ibtest Switch Operations 5-67
5.5.4 Interpreting the ibtest log files 5-68
5.6 Health Check and Baselining Tools 5-69
5.6.1 Usage Model 5-69
5.6.2 Common Operations and Options 5-70
5.6.3 fabric_analysis 5-72
5.6.4 chassis_analysis 5-76
5.6.5 hostsm_analysis 5-81
5.6.6 esm_analysis 5-83
5.6.7 all_analysis 5-86
5.6.8 Manual and Automated Usage 5-87

Section 6 MPI Sample Applications

6.1 OSU Latency 6-2
6.2 OSU Latency2 6-2
6.3 OSU Bandwidth 6-2
6.4 OSU Bandwidth2 6-3
6.5 OSU Bidirectional Bandwidth 6-3

6.6 High Performance Linpack (HPL) 6-3
6.7 Pallas 6-4

Appendix A Fast Fabric Quick Install Checklist

A.1 Setup The Fabric A-1
A.2 Installing and verifying Firmware on the IB Chassis A-2
A.3 Installing and Configuring the Subnet Manager A-2
A.4 Installing and Verifying Firmware on the IB Switches A-2
A.5 Install Infiniband on the Remaining Servers A-2
A.6 Verifying Infiniband on the Remaining Servers A-3
A.7 Complete Installation of additional IB Management Nodes A-3
A.8 Configure and initialize health check tools A-4

Appendix B Fast Fabric Configuration Files

B.1 fastfabric.conf B-1
B.2 iba_mon.conf B-6
B.3 Host List Files B-9
B.4 Chassis List Files B-9
B.5 Selection of slots within a chassis B-10
B.6 Switch List Files B-11
B.7 Port List Files B-12

Appendix C Configuration of IPoIB Name Mapping

Appendix D Multi-Subnet Fabrics

D.1 Primarily Independent Subnets D-1
D.2 Overlapping Subnets D-3

Section 1

Introduction

This manual describes installation, configuration and administration task information for the Fast Fabric Toolset.

This manual is organized as follows:

Section 1 describes the intended audience and technical support.

Section 2 describes the Fast Fabric Toolset.

Section 3 describes getting started with Fast Fabric.

Section 4 describes the Fast Fabric Textual User Interface (TUI) menu.

Section 5 describes the Fast Fabric command tools and test tools.

Section 6 describes MPI Sample Applications.

Appendix A presents the Fast Fabric Quick Install Checklist.

Appendix B describes the Fast Fabric Configuration Files.

Appendix C provides information on the configuration of IPoIB name mapping.

Appendix D provides information on configuring Multi-Subnet Fabrics.

1.1 Intended Audience

This manual is intended to provide network administrators and other qualified personnel a reference for installation, configuration and administration task information for the Fast Fabric toolset.

1.2 License Agreements

Refer to the *QLogic Software End User License Agreement* for a complete listing of all license agreements affecting this product.

1.3 Technical Support

Customers should contact their authorized maintenance provider for technical support of their QLogic products. QLogic-direct customers may contact QLogic Technical Support; others will be redirected to their authorized maintenance provider.

Visit the QLogic support Web site listed in Contact Information for the latest firmware and software updates.

1.3.1 Availability

QLogic Technical Support for products under warranty is available during local standard working hours excluding QLogic Observed Holidays.

1.3.2 Contact Information

Support Headquarters	QLogic Corporation 4601 Dean Lakes Blvd Shakopee, MN 55379 USA
QLogic Web Site	www.qlogic.com
Technical Support Web Site	support.qlogic.com
Technical Support Email	support@qlogic.com
Technical Training Email	tech.training@qlogic.com
North American Region	
Email	support@qlogic.com
Phone	+1-952-932-4040
Fax	+1 952-974-4910
All other regions of the world	
QLogic Web Site	www.qlogic.com

Section 2

Fast Fabric Overview

2.1 Feature Overview

The Fast Fabric Toolset is designed to both simplify and expedite common InfiniBand (IB) cluster management tasks. Fast Fabric can assist in generic management tasks as well as InfiniBand installation, upgrade, configuration and verification tasks.

Fast Fabric has the following key capabilities:

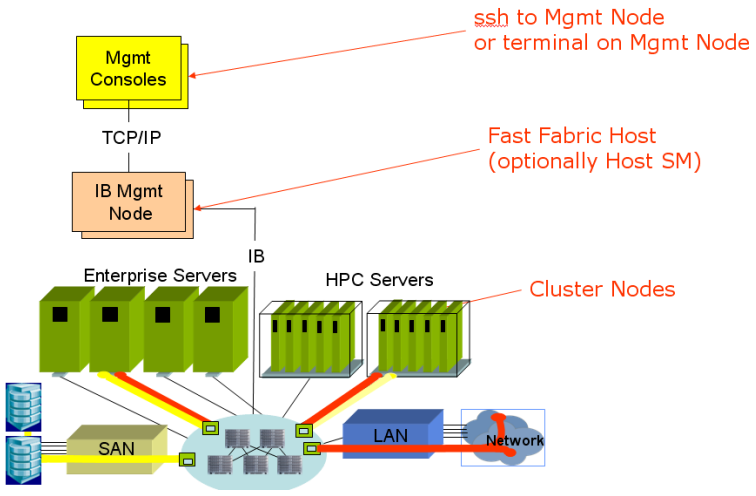
- Accelerates initial fabric installation
  - Verify host management network connectivity
  - Verify host OS levels
  - Sets up ssh keys
  - Performs initial InfiniBand software installation
  - Configures Internet Protocol over InfiniBand (IPoIB) IP addresses
  - Performs InfiniBand driver upgrades or the installation of additional InfiniBand drivers
- Verifies key fabric installation metrics:
  - Components in fabric
  - Link error counters
  - Link widths and speeds
  - IB and PCI bus bandwidth
  - IB end-to-end latency
  - IPoIB connectivity
  - Subnet Agent (SA) visibility of all nodes
  - IB connectivity of all switches and nodes
- Aids in diagnosis of fabric problems
  - Fabric error isolation
  - Fabric topology analysis
  - Fabric route analysis

- Aids in ongoing fabric status and configuration monitoring
  - Automated fabric health checks and configuration baseline compare
  - Automated chassis health checks and configuration baseline compare
  - Automated SM health checks and configuration baseline compare
- Provides tools to accelerate common host administration tasks
  - Executes commands across many hosts
  - Copies files to and from many hosts
  - Edits host-specific files across many hosts
- Provides tools to accelerate common chassis and switch administration tasks
  - Manage firmware levels on switches and chassis
  - Execute commands across many chassis
- Assists in the initial benchmarking and tuning of High Performance Computing (HPC) fabrics.

Fast Fabric includes both a Textual User Interface (TUI) menu system as well as command line tools. The TUI presents the menus in a typical order of execution for a new fabric install, hence simplifying fabric installation for new users. All operations available in the TUI can also be accomplished via the command line. The command line tools are designed to permit customer specific scripts to invoke the command line tools.

2.2 Fast Fabric Architecture

Figure 2-1. Fast Fabric Architecture



Fast Fabric is typically installed on one or more IB Management Nodes. The IB Management Node must be connected to the rest of the cluster via both InfiniBand and a management network. The management network may be the primary InfiniBand network (IPoIB) or Ethernet. The management network will be used for Fast Fabric host setup and administration tasks. It may also be used for other aspects of server administration or operation.

Depending on cluster size and design, the IB Management node may also be used as the master node for starting MPI jobs. It may also be used to run a QLogic Host SM and other management software. Consult the QLogic SM documentation for details and what combinations are valid.

Note: When InfiniBand is used as the management network, Fast Fabric will not be able to install host IB software nor configure IPoIB, however it will be able to support host IB software upgrades, verification and all the other features of Fast Fabric.

If remote access to Fast Fabric is desired, setup remote access to the IB Management Node via ssh, telnet, X windows, VNC or any other mechanism which will allow the remote user to access a Linux Command Line shell. Typically Fast Fabric is only used by cluster administrators.

2.2.1 How Fast Fabric Works

Fast Fabric consists of a variety of tools to administer hosts, chassis and externally managed switches. Depending on the tool, the method of accessing and administering the target devices may differ.

The following methods are used by Fast Fabric:

Table 2-1. Fast Fabric Methods

Method	Examples
Inband access via IB	Fabric topology reports, SA database queries, fabric error and link speed analysis, tools for externally managed switches, etc
Login via management network	Host setup and installation, tools for internally managed chassis, etc
MPI job startup (can be inband or via management network)	Verify MPI performance, running sample MPI benchmarks

Typically tools which login to other hosts will do such in a password-less manner using ssh or telnet (configurable). Tools which login to internally managed chassis can use ssh or telnet (configurable). Chassis tools can prompt for a single password for all chassis or can be preconfigured with the password. These approaches permit

the tools to operate with minimal user interaction and hence reduce the time to perform operations against many hosts or chassis.

After initial installation, Fast Fabric can be configured to use IPoIB instead of the management network.

NOTE: Any reconfigurations that affect IPoIB or involve installing new IB hosts will not be able to use IPoIB.

Section 3

Getting Started

Before using the Fast Fabric toolset, the Site Implementation Engineer must perform the tasks described in the sections which follow. To aid in keeping track of steps performed a checklist is provided (see appendix A). During the setup procedure, the Fast Fabric configuration files which must be edited or created are described throughout the procedure. For more information about the configuration files used by Fast Fabric see appendix B.

The instructions below describe the basic fabric installation and verification sequence for a typical single IB subnet fabric. For more information on installation and verification of multiple IB subnet fabrics, see appendix D.

Some of the tasks are only applicable when Linux is being used. They will be marked with **(Linux)**. Similarly some of the tasks are only applicable when QuickSilver Linux IB software is being used on the hosts. Those will be marked with **(Host)**. All tasks which are applicable only when SilverStorm IB Switches or SilverStorm IB Chassis are being used will be marked with **(Switch)**. All remaining tasks are generally applicable to all environments and will be marked with **(All)**.

NOTE: Some of the Linux steps may be applicable to other Unix-like operating systems if it is desired to enable use of non-IB specific Fast Fabric tools (such as cmdall) against the given hosts.

3.1 Design the Fabric

Prior to beginning the installation and setup of the fabric, its important to carefully design and plan the installation. Part of the design plan must include identification of which servers will be the administration nodes for the cluster and hence where Fast Fabric will be installed.

For large clusters, cable, power, and cooling plans are very important and must be carefully considered. These plans drive the ultimate layout of equipment in the racks. A typical configuration involves leaf switches and servers in the same racks, with core switches in centrally located racks. This minimizes both cable lengths and complexity. It is also recommended to place the IB switches at the bottom of a rack. This allows inter-rack cables to be cleanly routed below the floor (some sites use cable routing above the racks in which case placing the IB switches near the top of the rack is recommended).

NOTE: The overall physical design has many complex aspects, such as power, cooling and rack layout which are beyond the scope of this document.

3.2 Set Up the Fabric

1. **(All)** The first step in any installation is to physically install the hardware:
   - Servers
   - Core and leaf InfiniBand switches, such as the SilverStorm 9024 and 9000 Multi-Protocol Fabric Directors (9020, 9040, 9080, 9120 and 9240).
   - Virtual I/O systems, such as the EVIC and FVIC cards for the SilverStorm 9000 Multi-Protocol Fabric Directors Series.

NOTE: When installing externally managed switches (such as the SilverStorm 9024-FC switch), take note of the Node GUID. This is typically on a label on the case of the switch. The Node GUID will be needed later to configure and manage the switch(es).

2. **(All)** Within each server a host channel adapter (HCA), such as the QuickSilver HCA 7000 or 9000 must be installed. Refer to the *QuickSilver Fabric Access Quick Start Guide* for instructions.
3. **(All)** Prior to installing software, the hardware configuration should be reviewed to ensure everything was installed according to plan. Later during the installation Fast Fabric tools may also be used to help verify the installation.
4. **(Linux)** Install the desired Linux OS version (with the same kernel distribution) on all hosts. Generally the IB Management node(s) (i.e., the host which will run Fast Fabric) should have a full install and must include the Tcl, Expect and TCLx packages. If Redhat Enterprise Server 3 or later is being installed, only the Tcl and Expect packages are required.
   For MPI clusters install the C and Fortran compilers along with their associated tools on the IB Management node(s).

NOTE: All hosts must have a command-line prompt ending in "**#** " or "**$** ". Make certain there is a space after either "**#**" or "**$**". Such a prompt must be used for the root user as well as any other user codes the user intends Fast Fabric to make use of.

NOTE: To simplify the use of Fast Fabric to setup ssh security, it is recommended to install all servers with the same root password. If desired after ssh has been setup using Fast Fabric, the user may change the root passwords.

NOTE: Consult the QuickSilver Fabric Access Linux Host Release Notes for a list of supported OS versions.

5. **(Linux)** Enable remote login as root to each host:
   In order for Fast Fabric to manage the hosts, the IB Management Node must be able to securely login as root to each host. This can be accomplished using either ssh or rsh. SSH is recommended due to its higher level of security. If

ssh is used, no additional manual steps are require at this stage (typically Linux OS installation will enable ssh)
Alternatively, if its desired to use rsh during fabric installation and/or operation, the following steps must be performed on each node such that the IB Management Node can login using rsh as user root.

a. Each node must be configured such that the IB management node can **rsh** into it. The IB management node must also be able to **rsh** into itself. Typically this requires that a **.rhosts** file be created in **/root** such as:

```
<mgmthost name> root
<mgmthost name.domain name> root
localhost root
<mgmthost IP address>
```

where mgmthost is the network name of the IB Management Node and domain is the network domain name of the master. The .rhosts file must have permissions of 640. Also, rsh should be enabled on each node.

Enable **rsh** by editing the **/etc/xinetd.d/rsh** file and setting:

```
disable=no
```

This can also be accomplished using:

```
chkconfig rsh on
```

Also enable rexec and rlogin using the above steps.

b. Execute **mv /etc/securetty /etc/securetty.bak**

6. **(All)** TCP/IP Host Name resolution:
Fast Fabric and TCP/IP will need to resolve hostnames to Management Network and/or IPoIB IP addresses. If the management network is not IPoIB, each host will need both a management network name and an IPoIB network name. In which case, a recommended convention is to use the actual hostname as the management network name and <HOSTNAME>-ib as the IPoIB network name (where <HOSTNAME> is the management network name of the given host)
Typically name resolution is accomplished by configuring a DNS server on the management network with both management network and IPoIB addresses for each host (and QLogic internally managed IB chassis). Alternately a /etc/hosts file may be created on the IB Management node. Fast Fabric can then propagate this /etc/hosts file to all the other hosts.

If using the /etc/hosts approach:

On the master node, add all the Ethernet and IPoIB addresses into the **/etc/hosts** file. For the IPoIB convention, use **<HOSTNAME>-ib**. The **localhost** line should not be edited.

The **`/etc/hosts`** file should not have any node-specific data (the following section will step through the task of copying this file to all the nodes).

If using DNS:

Consult the documentation for the DNS server being used. Make sure to edit the /etc/resolv.conf configuration on the IB Management Node to use the proper DNS server. Consult the Linux OS documentation for more information on configuring /etc/resolv.conf. This file is typically configured during OS installation.

If /etc/resolv.conf must be manually configured for each host, Fast Fabric can aid in copying this to all the hosts. In which case, the /etc/resolv.conf file created on the IB Management Node must not have any node-specific data and must be appropriate for use on all hosts. A later section will step through the task of copying this file to all the nodes.

7. **(All)** NTP setup - it is recommended to configure an NTP server for the cluster and have all the hosts and Internally-Managed chassis synchronize their clocks with the NTP server. Consult the Linux OS documentation for information on how to configure NTP servers and clients.
8. **(All)** On the IB Management node, install the Fabric Access Software using the procedure documented in the *Fabric Access Software Users Guide*. The IB Management Node must have at least Fast Fabric, the IB Stack and IPoIB installed and configured. For MPI clusters running the QuickSilver Host stack, the IB Management Node should also include the MPI Runtime and MPI Development packages, and if the user desires to rebuild MPI itself, the IB Development package and MPI Source packages will also be required.

   After completing the install, reboot the IB Management node.

NOTE: When managing a cluster where compute nodes are not running the QuickSilver host stack or where the IPoIB settings on the compute nodes are incompatible with the IB Management node (for example when a 4K MTU is used on the compute nodes), it is recommended not to run IPoIB on the IB management nodes.

3.3 Using Fast Fabric

The initial installation and verification process is best performed using the Fast Fabric TUI menu system. The main menu can be invoked using the iba_config command. The main menu is as follows:

```
SilverStorm Technologies Inc. InfiniBand 4.1.1.0.15 Software

   1) Show Installed Software
   2) Reconfigure IP over IB
   3) Reconfigure Driver Autostart
   4) Update HCA Firmware
   5) Generate Supporting Information for Problem Report
   6) Host Setup via Fast Fabric
   7) Host Admin via Fast Fabric
   8) Chassis Admin via Fast Fabric
   9) Externally Managed Switch Admin via Fast Fabric
   a) Uninstall Software

   X) Exit
```

In the above menu, items 6-9 represent the Fast Fabric menus. The operation of this menu is the same as the INSTALL and iba_config functions documented in the *QuickSilver Fabric Access Users Guide*. Pressing a key 1-9 or a will invoke the given submenu. Pressing X will exit the menu system.

Selection of a Fast Fabric menu (6-9) will present a submenu similar to the following:

```
SilverStorm Technologies Inc. IB Host Setup Menu (4.1.1.0.15)
Fast Fabric Host List: /etc/sysconfig/iba/hosts

0) Edit Config and Select/Edit Hosts Files   [Perform]
1) Verify Hosts via Ethernet ping            [Perform]
2) Verify rsh/rcp Configured                 [ Skip  ]
3) Setup Password-less ssh/scp               [Perform]
4) Copy /etc/hosts to all hosts              [ Skip  ]
5) Show uname -a for all hosts               [Perform]
6) Install/Upgrade InfiniServ Software       [Perform]
7) Configure IPoIB IP Address                [Perform]
8) Build MPI Test Apps and Copy to Hosts     [Perform]
9) Reboot Hosts                              [Perform]
a) Refresh ssh Known Hosts                   [Perform]
b) Rebuild MPI Library and Tools             [ Skip  ]
c) Run a command on all hosts                [ Skip  ]
d) Copy a file to all hosts                  [ Skip  ]
e) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

The submenus typically present operations in the typical order they would be used during an installation. Pressing the keys corresponding to menu items (0-e in the example above) will toggle the Skip/Perform selection for the given item. As shown in the example above, more than 1 item may be selected. Once the desired set of

items have been selected, press P. To unselect all items, press N. Pressing X or ESC will exit this menu and return to the Main Menu.

If more than 1 item is selected, the items will be performed in the order shown in the menu. This is the typical order desired during fabric setup. If it's desired to perform items in a different order, select a single item and press P to perform it by itself. Then repeat. An opportunity will be presented after each item to abort:

```
Hit any key to continue (or ESC to abort)...
```

If ESC is pressed, the sequence of operations will be aborted and return to the previous menu. Any other key will result in the next selected menu item being performed. This prompt is also shown after the last selected item completes, hence permitting an opportunity to review the results before the screen is cleared to display the menu.

At the top of each Fast Fabric menu, the file listing the components to operate on is shown. For example:

```
Fast Fabric Host List: /etc/sysconfig/iba/hosts
```

On each Fast Fabric menu, item **0** will permit a different file to be selected and will permit the editing of the file (using the editor selected via the **EDITOR** environment variable). In addition it will also permit review and editing of the **fastfabric.conf** file. The **fastfabric.conf** file guides the overall configuration of Fast Fabric and describes cluster specific attributes of how Fast Fabric will operate. It is discussed in greater detail in appendix B.

During the execution of each menu selection, the actual Fast Fabric command line tool being used will be shown. This can be used as an educational aid to learn the tools.

3.4 Installing and Verifying Firmware on the SilverStorm IB Chassis

If the fabric contains SilverStorm 9000 series internally-managed IB switches, Fast Fabric may be used to aid the installation and configuration of the switches.

Prior to using Fast Fabric the following minimal steps need to be performed:

1. **(Switch)** Connect each SilverStorm chassis to the management network via its Ethernet management port. Chassis with redundant management should have both Ethernet management ports connected.
2. **(Switch)** Assign each SilverStorm chassis a unique IP address and appropriately configure the chassis Ethernet management port network settings.
3. **(Switch)** Select a unique name which will be used for each SilverStorm Chassis. This name should be configured in DNS or /etc/hosts as the TCP/IP name for the chassis Ethernet management port. In addition this should be configured as the IB Node Description for the chassis via the chassis GUI or CLI.

a. When Virtual I/O controllers (VIC) are installed in a chassis, each VIC should also be assigned a unique name.

4. **(Switch)** Configure the administrator password on each SilverStorm Chassis

NOTE: Newer versions of SilverStorm chassis firmware permit SSH keys to be configured within the chassis for secure password-less login. In this case it is recommended to configure SSH keys in the chassis at this point such that the IB Management Node can login as admin without a password.

NOTE: When using versions of SilverStorm chassis firmware that do not support SSH keys, to simplify the use of Fast Fabric it is recommended to install all chassis with the same admin password.

5. **(Switch)** Mount or copy the relevant chassis firmware CD(s) or files onto the Fast Fabric management node. During the steps below the *.pkg files on the CD will be used to upgrade the firmware on each chassis.

NOTE: When copying files its best to place all files at a given firmware level into a single directory whose name indicates the firmware revision number

Once the above steps have been completed, additional setup of the Chassis may be performed using Fast Fabric.

1. **(Switch)** Select the "Chassis Admin via Fast Fabric" option from the main menu.
2. **(Switch)** Select the items shown as "Perform" in the menu below and press the P key to perform them:

```
SilverStorm Technologies Inc. IB Chassis Admin Menu (4.1.1.0.15)
Fast Fabric Chassis List: /etc/sysconfig/iba/chassis

0) Edit Config and Select/Edit Chassis Files [Perform]
1) Verify Chassis via Ethernet ping          [Perform]
2) Update Chassis Firmware                   [Perform]
3) Show Status of Chassis IB Ports           [ Skip  ]
4) Reboot Chassis                            [ Skip  ]
5) Generate all Chassis Problem Report Info  [ Skip  ]
6) Run a command on all chassis              [ Skip  ]
7) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

3. **(Switch)** "Edit Config and Select/Edit Chassis Files" will permit the chassis and fastfabric.conf files to be edited. When placed in the editor for fastfabric.conf, review all the settings. Especially review the **`FF_CHASSIS_LOGIN_METHOD`**

and **FF_CHASSIS_ADMIN_PASSWORD**. Consult appendix B for more information about **fastfabric.conf**.

NOTE: Fast Fabric will provide the opportunity to enter the chassis password interactively when needed. Hence it's not necessary to place it within fastfabric.conf. If it is desired to instead keep the QLogic Chassis admin password in fastfabric.conf, its recommended to change the fastfabric.conf permissions to be 0x600 (eg. root only access).

NOTE: Newer versions of chassis firmware permit ssh keys to be configured within the chassis for secure password-less login. In which case there is no need to configure a FF_CHASSIS_ADMIN_PASSWORD and FF_CHASSIS_LOGIN_METHOD can be ssh. Consult the *SilverStorm 9000 Users Guide* for more information.

When placed in the editor for chassis, create the file with a list of the chassis names (the TCP/IP Ethernet management port names assigned above) or IP addresses (Use of names is recommended). One entry per line. Such as:

```
Chassis1
Chassis2
```

NOTE: Do not list externally managed switches, such as the SilverStorm 9024FC switches in this file. Those will be covered in the next section.

For further details about the file format refer to the section "Selection of Chassis" on page 5-4

4. **(Switch)** "Verify Chassis via Ethernet ping" will ping each selected chassis over the management network. If all chassis were found, continue to the next step. If some chassis were not found, abort out of the menu and review the following for those chassis which were not found:
   - Is chassis powered on and booted
   - Is chassis connected to management network
   - Is chassis IP address and network settings consistent with DNS or /etc/hosts
   - Is Management node connected to the management network
   - Are Management node IP address and network settings correct
   - Is the management network itself up (switches, routers, etc)
   - Is correct set of chassis listed in the chassis file (the previous step may be repeated to review and edit the file as needed)
5. **(Switch)** "Update Chassis Firmware" will permit the chassis firmware version to be verified and updated as needed.

NOTE: The chassis must be running firmware version 4.0.0.4.3 or later to perform this function. If the chassis is not up to this level, it will need to be manually updated via the chassis GUI. See the *SilverStorm 9000 Users Guide* for more information.

NOTE: Consult the relevant chassis firmware release notes to ensure any prerequisites for the upgrade to the new firmware level have been met prior to performing the upgrade via Fast Fabric.

When prompted:

```
Multiple Firmware files and/or Directories may be space
separated

Shell wildcards may be used

For Directories all .pkg files in the directory tree will be used

Enter Files/Directories to use (or none):
```

specify the directory where the relevant firmware files have been stored. This can be the mount point of the CD or the directory to which the files were copied in a previous step.

Since the fabric is not yet operational, it's recommended to answer "y" to:

```
Would you like to run the firmware now? [n]:
```

Fast Fabric will ensure that all chassis are running the firmware level provided and install and/or reboot each chassis as needed

If any chassis fails to be updated, use the "View ibtest result files" option to review the result files from the update. Refer to the section “Interpreting the ibtest log files” on page 5-68for more details.

6. **(Switch)** If there are any other operations which need to be performed on all chassis, they may be performed using the "Run a command on all chassis" option.Each time this is executed a single chassis CLI command may be specified to be executed against all selected chassis. Using such commands, additional setup or verification of the chassis may be performed.

3.5 Installing and Configuring the Subnet Manager

(All) At this point the subnet manager (SM) for the fabric must be installed or enabled. Consult the *QuickSilver Fabric Manager and Fabric Viewer Users Guide* for information on how to install, enable and configure the SM.

When using the QuickSilver host-based SM, a typical installation will place Fast Fabric and the host SM on the same IB Management Node. If desired, it is also valid to place Fast Fabric on its own independent management node, perhaps along with other 3rd party management applications (such as MPI job schedulers, etc).

The steps which follow will require that an SM be operational within the fabric.

3.6 Installing and Verifying Firmware on the IB Switches

If the fabric contains SilverStorm 9024FC series externally managed switches, Fast Fabric may be used to aid the installation and configuration of the switches.

Prior to using Fast Fabric the following minimal steps need to be performed:

1. **(Switch)** Select a unique name which will be used for each Switch. This name will be configured as the IB Node Description for the switch in the steps below.

NOTE: Externally managed switches do not have an Ethernet port and hence will not have a TCP/IP name.

2. **(Switch)** Mount or copy the relevant switch firmware CD(s) or files onto the Fast Fabric management node. During the steps below the *.emfw files on the CD will be used to upgrade the firmware on each switch.

NOTE: When copying files its best to place all files at a given firmware level into a single directory whose name indicates the firmware revision number

Once the above steps have been completed, additional setup of the switches may be performed using Fast Fabric.

3. **(Switch)** Select the "Externally Managed Switch Admin via Fast Fabric" option from the main menu.
4. **(Switch)** Select the items shown as "Perform" in the menu below and press the P key to perform them:

```
SilverStorm Technologies Inc. IB Switch Admin Menu (4.1.1.0.15)
Fast Fabric Externally Managed Switch List:
/etc/sysconfig/iba/ibnodes

0) Edit Config and Select/Edit Switch Files  [Perform]
1) Verify Switch via Firmware dump           [ Skip  ]
2) Update Switch Firmware                    [Perform]
3) Reboot Switch                             [ Skip  ]
4) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

5. **(Switch)** "Edit Config and Select/Edit Switch Files" will permit the ibnodes and fastfabric.conf files to be edited. When placed in the editor for fastfabric.conf,

review all the settings. Refer to appendix B for more information about fastfabric.conf.

When placed in the editor for ibnodes, create the file with a list of the switch node guids and desired switch names, one entry per line. Such as:

```
0x00066a00d9000138,edge1
0x00066a00d9000139,edge2
```

NOTE: Do not list Internally managed chassis, such as the SilverStorm 9000 chassis in this file. Those were covered in the previous section.

NOTE: If the IB path from the IB Management node to other switch nodes is through a 9024FC which is to be updated, the ibnodes file should omit that 9024FC switch at this time. Otherwise the reboot for the 9024FC in the path could disrupt the updates of other switches.

For further details about the file format, refer to section “Selection of Switches” on page 5-7

If needed, a SA query such as the following can be used to get a list of all switches, however this will include both internally and externally managed switches and hence the output must be edited to leave only the SilverStorm externally-managed switches:

```
saquery -t sw -o nodeguid
```

6. **(Switch)** "Update Switch Firmware" will permit the switch firmware version to be updated and the switch node name set.

NOTE: Consult the relevant switch firmware release notes to ensure any prerequisites for the upgrade to the new firmware level have been met prior to performing the upgrade via Fast Fabric.

When prompted:

```
Multiple Firmware files and/or Directories may be space
separated
Shell wildcards may be used
For Directories all .emfw files in the directory tree will be used
Enter Files/Directories to use (or none):
```

specify the directory where the relevant firmware files have been stored. This can be the mount point of the CD or the directory to which the files were copied in a previous step.

Since the fabric is not yet operational, it's recommended to answer "y" to:

```
Would you like to run the firmware now? [n]:
```

Fast Fabric will update the firmware on all switches and set the node names as per the ibnodes file created in a previous step. Each switch will then be rebooted.

If any switch fails to be updated, use the "View ibtest result files" option to review the result files from the update. Refer to the section “Interpreting the ibtest log files” on page 5-68 for more details.

If some switches were not found review the following for those switches which were not found:

- Is switch powered on
- Is switch connected to IB network
- Is Management node connected to the IB network
- Is the SM running on the IB network
- Is correct set of switches listed in the ibnodes file (the previous step may be repeated to review and edit the file as needed)

7. **(Switch)** If any 9024FC switches were skipped above in step 5 and 6, these steps should be repeated for those switches. In this case it is recommended to create a separate file with a name other than ibnodes. An alternate name may be specified at the prompt:

```
Select Switch File to Use/Edit
[/etc/sysconfig/iba/ibnodes]:
```

3.7 Installing InfiniBand on the Remaining Servers

Fast Fabric may now be used to install and configure the remaining hosts and verify overall operation of the fabric.

1. **(Linux)** Select the "Host Setup via Fast Fabric" option from the main menu.
2. Select the items shown as "Perform" in the menu below and press the P key to perform them:

```
SilverStorm Technologies Inc. IB Host Setup Menu (4.1.1.0.15)
Fast Fabric Host List: /etc/sysconfig/iba/hosts

0) Edit Config and Select/Edit Hosts Files   [Perform]
1) Verify Hosts via Ethernet ping            [Perform]
2) Verify rsh/rcp Configured                 [ Skip  ]
3) Setup Password-less ssh/scp               [Perform]
4) Copy /etc/hosts to all hosts              [ Skip  ]
5) Show uname -a for all hosts               [Perform]
6) Install/Upgrade InfiniServ Software       [Perform]
7) Configure IPoIB IP Address                [Perform]
8) Build MPI Test Apps and Copy to Hosts     [Perform]
9) Reboot Hosts                              [Perform]
a) Refresh ssh Known Hosts                   [ Skip  ]
b) Rebuild MPI Library and Tools             [ Skip  ]
c) Run a command on all hosts                [ Skip  ]
d) Copy a file to all hosts                  [ Skip  ]
e) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

NOTE: If passwordless root login via rsh is to be used during fabric setup and operation, also select "Verify rsh/rcp Configured". However it is instead recommended that ssh be used, in which case this step can be skipped.

NOTE: If /etc/hosts will be used for name resolution (as opposed to using DNS), also select "Copy /etc/hosts to all hosts"

3. **(All)** "Edit Config and Select/Edit Hosts Files" will permit the hosts and fastfabric.conf files to be edited. When placed in the editor for fastfabric.conf, review all the settings. Especially review the FF_IPOIB_SUFFIX, .ff_host_basename_to_ipoib, ff_host_basename, FF_IPOIB_NETMASK, FF_PRODUCT, FF_PACKAGES, FF_INSTALL_OPTIONS, FF_UPGRADE_OPTIONS and FF_ALL_ANALYSIS. Consult appendix B for more information about fastfabric.conf.

NOTE: During setup of passwordless ssh, Fast Fabric will provide the opportunity to enter the host root password interactively when needed. Therefore, it is recommended not to place it within fastfabric.conf. If it is desired to instead keep the root password for the hosts in fastfabric.conf, its recommended to change the fastfabric.conf permissions to be 0x600 (eg. root only access).

When placed in the editor for hosts, create the file with a list of the hosts names (the TCP/IP management network names) except the IB Management node from which Fast Fabric is presently being run, one entry per line. Such as:

```
host1
host2
```

NOTE: **Do not list the IB Management Node itself (i.e., the node where Fast Fabric is currently running).**

If multiple IB Management Nodes are to be used, they may be listed at this time and Fast Fabric can aid in their initial installation and verification.

For further details about the file format, refer to the section “Selection of Hosts” on page 5-3.

4. **(All)** "Verify Hosts via Ethernet ping" will ping each selected host over the management network. If all hosts were found, continue to the next step. If some hosts were not found, abort out of the menu and review the following for those hosts which were not found:
   - Is host powered on and booted
   - Is host connected to management network
   - Is host management network IP address and network settings consistent with DNS or /etc/hosts
   - Is Management node connected to the management network
   - Are Management node IP address and network settings correct
   - Is the management network itself up (switches, routers, etc)
   - Is correct set of hosts listed in the hosts file (the previous step may be repeated to review and edit the file as needed)
5. **(Linux)** "Verify rsh/rcp Configured" will confirm that passwordless rsh/rcp is properly configured such that the IB Management Node can access all the other hosts.

NOTE: It is recommended that ssh be used instead in which case this step may be skipped

6. **(Linux)** "Setup Password-less ssh/scp" will setup secure password-less ssh such that the IB Management Node can securely login to all the other hosts as root via the management network without requiring a password.

Password-less ssh is required by Fast Fabric, MPI test applications and most versions of MPI (including QuickSilver MPI).

7. **(Linux)** "Copy /etc/hosts to all hosts" will copy the /etc/hosts file on this host to all the other selected hosts.

NOTE: If DNS is being used, this step should be skipped.

NOTE: Typically, /etc/resolv.conf is setup as part of OS installation for each host. However, if /etc/resolv.conf was not setup on all the hosts during OS installation, the Fast Fabric "Copy a file to all hosts" operation could be done at this time to copy /etc/resolv.conf from the IB Management Node to all the other nodes.

8. **(Linux)** "Show uname -a for all hosts" will show the OS version on all the hosts. Review the results carefully to verify all the hosts have the expected OS version. In typical clusters all hosts will be running the same OS and kernel version.

If any hosts are identified with an incorrect OS version, the OS on those hosts should be corrected at this time and operation of this sequence should be aborted when prompted. As necessary all the preceding setup steps should then be repeated for those hosts (there is no harm in repeating them for all the hosts).

9. **(Host)** "Install/Upgrade InfiniServ Software" will install the IB software on all the hosts. By default it will look in the current directory for the $FF_PRODUCT.<VERSION>.tgz file. If it is not found in the current directory, it will prompt for input of a directory name where this file can be found.

When prompted, select to do an initial installation as follows:

```
Would you like to do an upgrade install? [y]: n
Would you like to do an initial install/load? [n]: y
```

NOTE: An initial installation will uninstall any existing InfiniServ software on the selected hosts. An upgrade install is not appropriate at this step.

If any hosts fail to be installed, use the "View ibtest result files" option to review the result files from the update. For more details, see “Interpreting the ibtest log files” on page 5-68.

10. **(Host)** "Configure IPoIB IP Address" will create the ifcfg-ib1 files on each host. The file will be created with a statically assigned IP address. The IPoIB IP address for each host will be determined by the resolver (Linux host command). If not found via the resolver, /etc/hosts on the given host will be checked.
11. **(Host)** "Build MPI Test Apps and Copy to Hosts" will build the MPI sample benchmarks on the IB Management Node and copy the resulting object files to all the hosts. This is in preparation for execution of MPI performance tests and benchmarks in a later step.
12. **(Linux)** "Reboot Hosts" will reboot all the selected hosts and ensure they go down and come back up (as verified via ping over the management network). When the hosts come back up, they will be running the IB software installed.
13. **(Linux)** If there are any other setup operations which need to be performed on all hosts, they may be performed using the "Run a command on all hosts" option.

Each time this is executed a Linux shell command (or sequence of commands separated by semicolons) may be specified to be executed against all selected hosts.

NOTE: It is recommended at this time to run the "date" command to verify the the date and time is consistent on all hosts. If needed "Copy a file to all hosts" may be used to copy the appropriate files to all hosts to enable and configure NTP.

3.8 Verifying InfiniBand on the Remaining Servers

Upon completion of the preceeding sections, the hosts are all booted, installed and operational. The subsequent steps will verify the operation of the hosts and fabric.

1. **(All)** Select the "Host Admin via Fast Fabric" option from the main menu.
2. Select the items shown as "Perform" in the menu below and press the P key to perform them:

```
SilverStorm Technologies Inc. IB Host Admin Menu (4.1.1.0.15)
Fast Fabric Host List: /etc/sysconfig/iba/allhosts

0) Edit Config and Select/Edit Hosts Files   [Perform]
1) Verify Hosts via Ethernet ping            [Perform]
2) Summary of Fabric Components              [Perform]
3) Show Status of Host IB Ports              [ Skip  ]
4) Verify Hosts see each other               [Perform]
5) Verify Hosts ping via IPoIB               [Perform]
6) Refresh ssh Known Hosts                   [Perform]
7) Check MPI Performance                     [Perform]
8) Generate all Hosts Problem Report Info    [ Skip  ]
9) Run a command on all hosts                [ Skip  ]
a) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

3. **(All)** "Edit Config and Select/Edit Hosts Files" will permit the hosts and fastfabric.conf files to be edited. When placed in the editor for fastfabric.conf, review all the settings. Especially review the FF_IPOIB_SUFFIX, ff_host_basename_to_ipoib, and ff_host_basename. Consult appendix B for more information about fastfabric.conf.

When placed in the editor for allhosts, create the file with the IB Management node's hosts names (the TCP/IP management network names) (shown as

mgmthost below for example) and include the hosts file previously created, one entry per line. Such as:

```
mgmthost
include /etc/sysconfig/iba/hosts
```

For further details about the file format refer to section “Selection of Hosts” on page 5-3.

4. **(All)** "Verify Hosts via Ethernet ping" will ping each selected host over the management network. If all hosts were found, continue to the next step. If some hosts were not found, abort out of the menu and review the following for those hosts which were not found:
   - Is the host powered on and booted
   - Is the host connected to management network
   - Is the host management network IP address and network settings consistent with DNS or /etc/hosts
   - Is the Management node connected to the management network
   - Are the Management node IP address and network settings correct
   - Is the management network itself up (switches, routers, etc)
   - Is the correct set of hosts listed in the hosts file (the previous step may be repeated to review and edit the file as needed)
5. **(All)** "Summary of Fabric Components" will provide a brief summary of the counts of components in the fabric including how many switch chips, hosts, and links are in the fabric. It will also indicate if any 1x links were found (which could indicate a poorly seated or bad cable). Review the results against the expected configuration of the cluster.

NOTE: The link count includes some internal links within the switch boxes. This means that the count displayed will be greater than the actual number of cables.

If components are missing or 1x links are found, they should be corrected. Subsequent steps will aid in locating any 1x links.

6. **(Host)** If desired "Show Status of Host IB Ports" will allow the state and symbol error counts of all ports to be manually reviewed.

   Instead it is recommended to run:

   ```
   iba_report -i 10 -o errors -o slowlinks
   ```

   on the IB Management node. This will check all the ports in the fabric for any links which have high error rates or are running at a lower speed than expected. Any such identified links should be diagnosed and corrected.

7. **(Host)** "Verify Hosts see each other" will verify that each host can see all the others via queries to the Subnet Administrator and the SA replica on each host has been fully populated.
8. **(Host)** "Verify Hosts ping via IPoIB" will verify that IPoIB is properly configured and running on all the hosts. This is accomplished via the IB management node pinging each host via IPoIB.
9. **(Linux)** "Refresh ssh Known Hosts" will refesh the ssh knowhosts file on the IB management node to include the IPoIB hostnames of all the hosts
10. **(Host)** "Check MPI Performance" will do a quick check of PCI and MPI performance.

    This displays the MPI latency and bandwidth between pairs of hosts (1-2, 3-4, 5-6, etc). The results are also written to the test.res file which may be viewed via the "View ibtest result files". Refer to the section “Interpreting the ibtest log files” on page 5-68 for more details.

    The numbers reported should be checked against the practical PCI speeds in the Performance Impact table below. If any pairs are not in the expected performance range, carefully examine the two hosts involved to verify that the PCI slot used, BIOS settings and any motherboard jumpers related to devices on PCI buses or slot speeds. Also verify HCA and riser cards are properly seated.

Table 3-1. Performance Impact

PCI Speed	Theoretical Max	Practical Bandwidth
133	1024MB/sec	800-900 MB/sec
100	770MB/sec	600-680 MB/sec
66	512MB/sec	400-450 MB/sec

3.9 Complete Installation of additional IB Management Nodes

If the fabric is to have more than one IB Management Node, the setup of the additional management nodes may be completed now. The previous steps will have performed basic software installation, setup and verification on those nodes. Now the management software itself must be installed and configured.

NOTE: The steps below assume a symmetrical configuration where all IB management nodes have the same connectivity and capabilities. In assymetrical configurations where the IB management nodes are not all connected to the same set of management networks and IB subnets, the files copied to each management node may need to be slightly different. For example configuration files for fabric_analysis may indicate different port numbers, or host files used for FastFabric and MPI may need to list different hosts. For multiple subnet configurations, refer to “Multi-Subnet Fabrics” on page D-1.

Repeat the following steps on each additional IB Management Node:

1. **(All)** Install the additional Fabric Access Software components using the procedure documented in the Fabric Access Software Users Guide. The IB Management Node must have at least Fast Fabric, the IB Stack and IPoIB installed and configured. For MPI clusters the IB Management Node should also include the MPI Runtime and MPI Development packages, and if the user desires to rebuild MPI itself, the IB Development package and MPI Source packages will also be required.

NOTE: Do not uninstall or replace existing configuration files which were previous created, especially IPoIB related configuration files.

2. **(All)** Copy the Fast Fabric configuration files from the initial IB Management Node. At least the following files should be copied:

```
/etc/sysconfig/fastfabric.conf
/etc/sysconfig/iba/hosts
/etc/sysconfig/iba/allhosts
/etc/sysconfig/iba/ibnodes
/etc/sysconfig/iba/chassis
```

   After copying the files, edit the hosts and allhosts files such that the file on each IB Management Node omits itself from the hosts files (but lists all other IB Management Nodes) and specifies itself in the allhosts file.

   See appendix B for a complete list of Fast Fabric configuration files.

3. **(Linux)** Perform "Setup Password-less ssh/scp" in the "Host Setup via Fast Fabric" menu and "Refresh ssh Known Hosts" in the "Host Admin via Fast Fabric" menu.

3.10 Configure and Initialize Health Check Tools

For more information on the health check tools, see the detailed discussion in “Health Check and Baselining Tools” on page 5-69. The Health check tools may be run on one or more IB management nodes within the cluster. This procedure should be followed on each IB management node from which the health check tools will be used.

1. **(All)** Edit fastfabric,conf and review the following parameters: FF_ANALYSIS_DIR, FF_ALL_ANALYSIS, FF_FABRIC_HEALTH, FF_CHASSIS_CMDS,_FF_CHASSIS_HEALTH, and FF_ESM_CMDS. FF_ALL_ANALYSIS should be updated to reflect the type of SM (esm or hostsm).
2. **(All)** If using Embedded SM(s) in QLogic IB Chassis, create /etc/sysconfig/iba/esm_chassis listing the chassis which are running SMs.

Create the file with a list of the chassis names (the TCP/IP Ethernet management port names assigned above) or IP addresses (Use of names is recommended). One entry per line. Such as:

```
Chassis1
Chassis2
```

For further details about the file format refer to the section “Selection of Chassis” on page 5-4.

3. **(All)** Perform a health check using: all_analysis -e. If any errors are encountered resolve the errors and rerun all_analysis -e until a clean run occurs.
4. **(All)** Create a cluster configuration baseline using: all_analysis -b
5. **(All)** If desired, schedule regular runs of all_analysis via cron or other mechanisms. Consult the Linux OS documentation for more information on cron. Also consult the section “Health Check and Baselining Tools” on page 5-69 for more information about all_analysis and its automated use.

3.11 Running HPL

As part of the installation process, a set of common MPI benchmarks have been installed. One of the more popular measures of overall performance is HPL. This is the application used to rate systems on the Top 500 list. The steps allow some initial runs of HPL to be made and provide some initial baseline numbers. The defaults provided should perform within 10%-20% of optimal HPL results for the cluster. Tuning for that additional 10%-20% is beyond the scope of this document.

1. **(Host)** To run HPL, first select a configuration file appropriate to your cluster. It is best to start with a small configuration to verify HPL has been properly compiled:

   a. cd /opt/iba/src/mpi_apps

   b. /config_hpl 2t

   will configure a two process test run of HPL.

2. **(Host)** Now create the file /opt/iba/src/mpi_apps/mpi_hosts listing the host names of all the hosts. Depending of your selection of VIADEV_PATH_METHOD in /opt/iba/src/mpi_apps/mpi.param.hpl the user can specify Ethernet or IPoIB host names. The default config will allow either.
3. **(Host)** Now run HPL:

   ```
   ./run_hpl 2
   ```

   Since this is a very small problem size the performance of the run will be much lower than the potential of the machine. So do not worry about performance, just whether or not the run was successful.

At this point the user is ready to move onto full scale HPL runs. Assorted sample HPL.dat files are provided in /opt/iba/src/mpi_apps/hpl-config. These files are a good starting point for most clusters and should get within 10-20% of the optimal performance for the cluster. The problem sizes used assume a cluster with 1GB of physical memory per processor (e.g., for a 2 processor node, 2GB of node memory is assumed). For each cluster size, 4 files are provided:

t - a very small test run (5000 problem size)

s - a small problem size on the low end of optimal problem sizes

m - a medium problem size

l - a large problem size

These can be selected using config_hpl. The following command displays the preconfigured problem sizes available:

```
./config_hpl
```

For example, to do a small run for a 256 processor cluster (i.e., 128 nodes of dual CPU systems):

1. Type ./config_hpl 256s and press Enter.
2. Type ./run_hpl 256 and press Enter.

During these runs the user should use top on a node to monitor memory and CPU usage. The xhpl should use 98-99% of the CPU. If any other processes are taking more than 1-2%, review the host configuration and stop these extra processes if possible. HPL is very sensitive to swapping. If a lot of swapping is seen, and xhpl is dropping below 97% for long durations, this may indicate a problem size that is too large for the memory and OS configuration.

At this point the user can continue to tune HPL to refine performance. Parameters in /opt/iba/src/mpi_apps/mpi.param.hpl and in HPL.dat can all affect HPL performance. In addition the selection of compiler and BLAS Math library may also significantly affect performance. The new HPL.dat files may be placed in /opt/iba/src/mpi_apps/hpl-config and use config_hpl to select them and copy them to all nodes in the run. Alternately scpall may be used to copy the file to all nodes. Refer to the section “Basic Setup and Administration Tools” on page 5-11 for more information on scpall.

3.12 Upgrading IB software

If an existing cluster which has been installed and verified needs to have Infiniband software upgraded, the following steps may be followed.

1. **(All)** On each IB Management Node, perform an upgrade installation of the Fabric Access Software using the procedure documented in the *Fabric Access Software Users Guide*. Each IB Management Node must have at least Fast Fabric, the IB Stack and IPoIB installed and configured. For MPI clusters the

IB Management Nodes should also include the MPI Runtime and MPI Development packages, and if the user desires to rebuild MPI itself, the IB Development package and MPI Source packages will also be required.

After completing the install, reboot each of the IB Management Nodes to ensure they are running the new IB software.

NOTE: Ensure that existing configuration is appropriately upgraded, especially Fast Fabric and IPoIB related configuration files. Consult the Fabric Access Software Users Guide and release notes for further information.

2. **(All)** Select the "Host Setup via Fast Fabric" option from the main menu.
3. Select the items shown as "Perform" in the menu below and press the P key to perform them:

```
SilverStorm Technologies Inc. IB Host Setup Menu (4.1.1.0.15)
Fast Fabric Host List: /etc/sysconfig/iba/hosts

0) Edit Config and Select/Edit Hosts Files   [Perform]
1) Verify Hosts via Ethernet ping            [ Skip  ]
2) Verify rsh/rcp Configured                 [ Skip  ]
3) Setup Password-less ssh/scp               [ Skip  ]
4) Copy /etc/hosts to all hosts              [ Skip  ]
5) Show uname -a for all hosts               [ Skip  ]
6) Install/Upgrade InfiniServ Software       [Perform]
7) Configure IPoIB IP Address                [ Skip  ]
8) Build MPI Test Apps and Copy to Hosts     [ Skip  ]
9) Reboot Hosts                              [Perform]
a) Refresh ssh Known Hosts                   [ Skip  ]
b) Rebuild MPI Library and Tools             [ Skip  ]
c) Run a command on all hosts                [ Skip  ]
d) Copy a file to all hosts                  [ Skip  ]
e) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

4. **(All)** "Edit Config and Select/Edit Hosts Files" will permit the hosts and fastfabric.conf files to be edited. When placed in the editor for fastfabric.conf, review all the settings. Especially review the FF_PRODUCT, FF_PACKAGES, and FF_UPGRADE_OPTIONS. See appendix B for more information about fastfabric.conf.

   Select a hosts list file which lists all the hosts except the IB Management nodes. If necessary create a new file at this time, potentially based on the existing /etc/sysconfig/iba/hosts file.

NOTE: **Do not list any of IB Management Nodes (eg. The nodes which have fast fabric installed)**

NOTE: The file may list the Management Network or IPoIB hostnames for the selected hosts

5. **(Host)** "Install/Upgrade InfiniServ Software" will upgrade the IB software on all the selected hosts. By default it will look in the current directory for the $FF_PRODUCT.<VERSION>.tgz file. If it is not found in the current directory, it will prompt for input of a directory name where this file can be found.

   When prompted, select to do an upgrade installation as follows:

   Would you like to do an upgrade install? [y]: y

NOTE: An upgrade installation will update any existing InfiniServ software on the selected hosts. An upgrade install is only valid for hosts which already have a previous version of InfiniServ software installed.

   If any hosts fail to be updated, use the "View ibtest result files" option to review the result files from the update. See the section “Interpreting the ibtest log files” on page 5-68 for more details.

6. **(Linux)** If there are any other setup operations which need to be performed on all hosts, they may be performed using the "Run a command on all hosts" option. Each time this is executed a Linux shell command (or sequence of commands separated by semicolons) may be specified to be executed against all selected hosts.

NOTE: NOTE: Check the relevant release notes for the new InfiniServ release being installed for any such additional required steps.

7. **(Linux)** "Reboot Hosts" will reboot all the selected hosts and ensure they go down and come back up (as verified via ping over the management network). When the hosts come back up, they will be running the IB software installed.
8. Repeat the verification steps for the fabric as discussed in the section “Verifying InfiniBand on the Remaining Servers” on page 3-16.

.

Section 4

Fast Fabric TUI Menu

Fast Fabric is easiest to use from the textual user interface (TUI) menu system. The menu system provides a way to perform all common tasks and presents common options. Additional less common options are available directly via the Command Line Tools documented in the next section.

In the sections that follow, the menu system will be discussed. The majority of menu items directly invoke various Fast Fabric command tools. As such the section on each menu item will indicate what command tool it invokes and a summary of the operation performed. For further details about the given command tool, consult the relevant section within "Basic Setup and Administration Tools" on page 5-11.

Some of the menu items are only applicable when Linux is being used. They will be marked with **(Linux)**. Similarly some of the menu items are only applicable when QuickSilver Linux IB software is being used on the hosts. Those will be marked with **(Host)**. All menu items which are applicable only when SilverStorm IB Switches or Chassis are being used will be marked with **(Switch)**. All remaining menu items are generally applicable to all environments and will be marked with **(All)**.

NOTE: Some of the Linux menu items may be applicable to other Unix-like operating systems if it is desired to enable the use of non-IB specific Fast Fabric tools (such as cmdall) against the given hosts.

The main menu can be invoked using the iba_config command. The main menu is as follows:

```
SilverStorm Technologies Inc. InfiniBand 4.1.1.0.15 Software

   1) Show Installed Software
   2) Reconfigure IP over IB
   3) Reconfigure Driver Autostart
   4) Update HCA Firmware
   5) Generate Supporting Information for Problem Report
   6) Host Setup via Fast Fabric
   7) Host Admin via Fast Fabric
   8) Chassis Admin via Fast Fabric
   9) Externally Managed Switch Admin via Fast Fabric
   a) Uninstall Software

   X) Exit
```

In the above menu, items 6-9 represent the Fast Fabric menus. The operation of this menu is the same as the INSTALL and iba_config functions documented in the

QuickSilver Fabric Access Software Users Guide. Selecting items 1-9 will display the given submenu. Pressing X will exit the menu system.

Selection of a Fast Fabric menu (6-9) will present a submenu such as below:

```
SilverStorm Technologies Inc. IB Host Setup Menu (4.1.1.0.15)
Fast Fabric Host List: /etc/sysconfig/iba/hosts

0) Edit Config and Select/Edit Hosts Files   [Perform]
1) Verify Hosts via Ethernet ping            [Perform]
2) Verify rsh/rcp Configured                 [ Skip  ]
3) Setup Password-less ssh/scp               [Perform]
4) Copy /etc/hosts to all hosts              [ Skip  ]
5) Show uname -a for all hosts               [Perform]
6) Install/Upgrade InfiniServ Software       [Perform]
7) Configure IPoIB IP Address                [Perform]
8) Build MPI Test Apps and Copy to Hosts     [Perform]
9) Reboot Hosts                              [Perform]
a) Refresh ssh Known Hosts                   [Perform]
b) Rebuild MPI Library and Tools             [ Skip  ]
c) Run a command on all hosts                [ Skip  ]
d) Copy a file to all hosts                  [ Skip  ]
e) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

The submenus typically present operations in the typical order they would be used during an installation. Pressing the keys corresponding to menu items (0-9, a-e in the example above) will toggle the Skip/Perform selection for the given item. As shown in the example above, more than 1 item may be selected. Once the desired set of items have been selected, press P. To unselect all items, press N. Pressing X or ESC will exit this menu and return to the Main Menu.

If more than 1 item is selected, the items will be performed in the order shown in the menu. This is the typical order desired during fabric setup. If it's desired to perform items in a different order, select a single item and press P to perform it by itself. Then repeat. An opportunity will be presented after each item to abort:

```
Hit any key to continue (or ESC to abort)...
```

If ESC is pressed, the sequence of operations will be aborted and return to the previous menu. Any other key will result in the next selected menu item being performed. This prompt is also shown after the last selected item completes, hence permitting an opportunity to review the results before the screen is cleared to display the menu.

At the top of each Fast Fabric menu, the file listing the components to operate on is shown. For example:

```
Fast Fabric Host List: /etc/sysconfig/iba/hosts
```

On each Fast Fabric menu, item 0 will permit a different file to be selected and will permit the editing of the file (using the editor selected via the EDITOR environment variable). In addition it will also permit review and editing of the fastfabric.conf file. The fastfabric.conf file guides the overall configuration of Fast Fabric and describes cluster specific attributes of how Fast Fabric will operate. It is discussed in greater detail in appendix B.

During the execution of each menu selection, the actual Fast Fabric command line tool being used will be shown. This can be used as an educational aid to learn the tools.

4.1 Host Setup via Fast Fabric

This menu is focused on initial host setup and installation of IB software on all the hosts.

```
0) Edit Config and Select/Edit Hosts Files   [ Skip  ]
1) Verify Hosts via Ethernet ping            [ Skip  ]
2) Verify rsh/rcp Configured                 [ Skip  ]
3) Setup Password-less ssh/scp               [ Skip  ]
4) Copy /etc/hosts to all hosts              [ Skip  ]
5) Show uname -a for all hosts               [ Skip  ]
6) Install/Upgrade InfiniServ Software       [ Skip  ]
7) Configure IPoIB IP Address                [ Skip  ]
8) Build MPI Test Apps and Copy to Hosts     [ Skip  ]
9) Reboot Hosts                              [ Skip  ]
a) Refresh ssh Known Hosts                   [ Skip  ]
b) Rebuild MPI Library and Tools             [ Skip  ]
c) Run a command on all hosts                [ Skip  ]
d) Copy a file to all hosts                  [ Skip  ]
e) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

4.1.1 Edit Configuration and Select/Edit Hosts Files

(All) This will permit the hosts and fastfabric.conf files to be edited. The hosts file selected and created via this menu should not list the Fast Fabric host itself. After editing the two files, an opportunity is given to edit them again or continue forward.

```
Selected Host File: /etc/sysconfig/iba/hosts

Do you want to edit/review/change the files? [y]:
```

The default will repeat the editing process, answer "n" to proceed to continue forward.

Refer to the section “Selection of Hosts” on page 5-3 for more details about the format of the hosts file.

4.1.2 Verify Hosts via Ethernet ping

(All) This will run the `pingall` command. All the hosts listed will be pinged via the Management Network.

4.1.3 Verify RSH/RCP Configured

(Linux) This will run the `check_rsh` command. This will confirm that passwordless rsh/rcp is properly configured such that the IB Management Node can access all the other hosts.

NOTE: It is recommended that SSH be used in place of the `check_rsh` command.

4.1.4
Setup Password-less SSH/SCP

(Linux) This will run the `setup_ssh -i ""` command. This will setup secure password-less SSH such that the IB Management Node can securely login to all the other hosts as root via the management network without requiring a password.

Password-less SSH is required by Fast Fabric, MPI test applications and most versions of MPI (including QuickSilver MPI).

4.1.5
Copy /etc/hosts to all hosts

(Linux) This will run the scpall /etc/hosts /etc/hosts command to copy the /etc/hosts file on this host to all the other selected hosts. This is not necessary when using a DNS server to resolve hostnames for the cluster.

4.1.6
Show uname -a for all hosts

(Linux) This run the `cmdall "uname -a"` command to show the OS version on all the hosts. Review the results carefully to verify all the hosts have the expected OS version. In typical clusters all hosts will be running the same OS and kernel version.

4.1.7
Install/Upgrade QuickSilver Software

(Host) This will run the `ibtest load` or `ibtest update` command to install the IB software on all the hosts. By default it will look in the current directory for the `$FF_PRODUCT.<VERSION>.tgz` file. If it is not found in the current directory, it will prompt for input of a directory name where this file can be found.

Prompts will guide the user through options:

- upgrade - updates all servers with new release. Only components previously installed are upgraded. Will fail for any hosts which have no InfiniServ IB software installed
- initial install/load - uninstalls any existing InfiniServ IB software and installs the given release based on fastfabric.conf installation options specified.

After the install is completed, the hosts will still need to be rebooted to bring up the new IB drivers. This can be performed using the "Reboot Hosts" menu item discussed below.

If any hosts fail to be updated, use the `View ibtest result files` option to review the result files from the update. For more details, see “Interpreting the ibtest log files” on page 5-68.

4.1.8
Configure IPoIB IP Address

(Host) This will run the ibtest configipoib command to create the `ifcfg-ib1` files on each host. The file will be created with a statically assigned IP address. The IPoIB IP address for each host will be determined by the resolver (Linux **`host`** command). If not found via the resolver, `/etc/hosts` on the given host will be checked.

4.1.9
Build MPI Test Apps and Copy to Hosts

(Host) This will build the MPI sample benchmarks on the IB Management Node and copy the resulting object files to all the hosts. This is in preparation for execution of MPI performance tests and benchmarks in a later step.

4.1.10
Reboot Hosts

(Linux) This will run the `ibtest reboot` command to reboot all the selected hosts and ensure they go down and come back up (as verified via ping over the management network). When the hosts come back up, they will be running the IB software installed.

4.1.11
Refresh SSH Known Hosts

(Linux) This will run the `setup_ssh -C -i""` command to refresh the ssh known hosts list on this server for the Management Network. This may be used to update security for this host if hosts are replaced, reinstalled, renamed, or repaired.

4.1.12
Rebuild MPI Library and Tools

(Host) This will rebuild the InfiniServ MPI Library itself and related tools (such as mpirun). This will be performed via the dobuild tool supplied with the InfiniServ MPI Source. Consult the *QuickSilver Fabric Access Software Users Guide* for more information.

4.1.13
Run a command on all hosts

(Linux) This will run the `cmdall` command. A Linux shell command (or sequence of commands separated by semicolons) may be specified to be executed against all selected hosts.

4.1.14
Copy a file to all hosts

(Linux) This will run the `scpall` command. A file on the local host may be specified to be copied to all selected hosts.

4.1.15
View ibtest result files

(All) This permits viewing of the `test.log` and `test.res` files that reflect the results from ibtest runs (such as for installing QuickSilver software or rebooting all hosts per menu items above). The user is also given the option to remove these files after viewing them.

If not removed, subsequent runs of `ibtest` from within the current directory will continue to append to these files.

4.2 Host Admin via Fast Fabric

This menu is focused on verifying hosts and the fabric as well as administration of all the hosts.

```
SilverStorm Technologies Inc. IB Host Admin Menu (4.1.1.0.15)
Fast Fabric Host List: /etc/sysconfig/iba/allhosts

0) Edit Config and Select/Edit Hosts Files   [ Skip  ]
1) Verify Hosts via Ethernet ping            [ Skip  ]
2) Summary of Fabric Components              [ Skip  ]
3) Show Status of Host IB Ports              [ Skip  ]
4) Verify Hosts see each other               [ Skip  ]
5) Verify Hosts ping via IPoIB               [ Skip  ]
6) Refresh ssh Known Hosts                   [ Skip  ]
7) Check MPI Performance                     [ Skip  ]
8) Generate all Hosts Problem Report Info    [ Skip  ]
9) Run a command on all hosts                [ Skip  ]
a) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

4.2.1 Edit Config and Select/Edit Hosts Files

All This will permit the allhosts and fastfabric.conf files to be edited. The `allhosts` file selected and created via this menu should list the Fast Fabric host itself. After editing the two files, an opportunity is given to edit them again or continue forward.

```
Selected Host File: /etc/sysconfig/iba/allhosts

Do you want to edit/review/change the files? [y]:
```

The default will repeat the editing process, answer `n` to roceed to continue forward.

Refer to the section “Selection of Hosts” on page 5-3 for more details about the format of the allhosts file.

4.2.2 Verify Hosts via Ethernet Ping

(All) This will run the `pingall` command. All the hosts listed will be pinged via the Management Network.

4.2.3 Summary of Fabric Components

(All) This will run the `fabric_info` command to provide a brief summary of the counts of components in the fabric including how many switch chips, hosts, and links are in the fabric. It will also indicate if any 1x links were found (that could indicate a poorly seated or bad cable). Review the results against the expected configuration of the cluster.

NOTE: The link count includes some internal links within the switch boxes. This means that the count displayed will be greater than the actual number of cables.

4.2.4 Show Status of Host IB Ports

(Host) This will run the showallports command to allow the state and symbol error counts of all host ports to be manually reviewed.

(All) Instead it is recommended to run:

```
iba_report -i 10 -o errors -o slowlinks
```

on the IB Management node. This will check all the ports in the fabric for any links which have high error rates or are running at a lower speed than expected. Any such identified links should be diagnosed and corrected.

4.2.5 Verify Hosts see each other

(Host) This will run the `ibtest sacache` command to verify that each host can see all the others via queries to the Subnet Administrator and the SA replica on each host has been fully populated.

4.2.6 Verify Hosts ping via IPoIB

(Host) This will run the `ibtest ipoibping` command to verify that IPoIB is properly configured and running on all the hosts. This is accomplished via the IB management node pinging each host via IPoIB.

4.2.7 Refresh SSH Known Hosts

(Linux) This will run the `setup_ssh -C` command to refresh the SSH known hosts list on this server for the IPoIB and Management Networks. This may be used to update security for this host if hosts are replaced, reinstalled, renamed, or repaired.

4.2.8
Check MPI Performance

(Host) This will run the `ibtest mpiperf` command to do a quick check of PCI and MPI performance.

This displays the MPI latency and bandwidth between pairs of hosts (1-2, 3-4, 5-6, etc). The results are also written to the test.res file which may be viewed via the `View ibtest result files`. Refer to the section “Interpreting the ibtest log files” on page 5-68 for more details.

The numbers reported should be checked against the practical PCI speeds in the Performance Impact section. If any pairs are not in the expected performance range, carefully examine the two hosts involved to verify that the PCI slot used, BIOS settings and any motherboard jumpers related to devices on PCI buses or slot speeds. Also verify HCA and riser cards are properly seated.

4.2.9
Generate all Hosts Problem Report Info

(Host) This will run the `captureall` command to collect configuration and status information from all hosts and generate a single `*.tgz` file which can be sent to the Support Representative.

4.2.10
Run a command on all hosts

(Linux) This will run the `cmdall` command. A Linux shell command (or sequence of commands separated by semicolons) may be specified to be executed against all selected hosts.

4.2.11
View ibtest result files

(All) This permits viewing of the `test.log` and `test.res` files which reflect the results from `ibtest` runs (such as those for installing QuickSilver software or rebooting all hosts per menu items above). The user is also given the option to remove these files after viewing them.

If not removed, subsequent runs of `ibtest` from within the current directory will continue to append to these files.

4.3
QLogic IB Chassis Admin via Fast Fabric

This menu is focused on administration of QLogic 9000 series internally managed IB chassis.

```
SilverStorm Technologies Inc. IB Chassis Admin Menu (4.1.1.0.15)
Fast Fabric Chassis List: /etc/sysconfig/iba/chassis

0) Edit Config and Select/Edit Chassis Files [ Skip  ]
1) Verify Chassis via Ethernet ping          [ Skip  ]
2) Update Chassis Firmware                   [ Skip  ]
3) Show Status of Chassis IB Ports           [ Skip  ]
4) Reboot Chassis                            [ Skip  ]
5) Generate all Chassis Problem Report Info  [ Skip  ]
6) Run a command on all chassis              [ Skip  ]
7) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

4.3.1
Edit the Configuration and Select/Edit Chassis Files

(Switch) This will permit the `chassis` and `fastfabric.conf` files to be edited. The chassis file selected and created via this menu should not list the Fast Fabric host itself. After editing the two files, an opportunity is given to edit them again or continue forward.

```
Selected Chassis File: /etc/sysconfig/iba/chassis
Do you want to edit/review/change the files? [y]:
```

The default will repeat the editing process, answer `n` to proceed to continue forward.

Refer to section “Selection of Chassis” on page 5-4 for more details about the format of the chassis file.

4.3.2
Verify Chassis via Ethernet Ping

(Switch) This will run the `pingall -C` command to ping each selected chassis over the management network.

4.3.3
Update Chassis Firmware

(Switch) This will run the **`ibtest -C update`** command to permit the chassis firmware version to be verified and updated as needed.

NOTE: The chassis must be running firmware version 4.0.0.4.3 or later to perform this function. If the chassis is not up to this level, it will need to be manually updated via the chassis GUI. See the *SilverStorm 9000 Users Guide* for more information.

NOTE: Consult the relevant chassis firmware release notes to ensure any prerequisites for the upgrade to the new firmware level have been met prior to performing the upgrade via Fast Fabric.

Prompts will guide the user through options:

- push - push firmware to each chassis but do not change selected nor running firmware
- select - push firmware to each chassis and select it for use on next reboot
- run - push firmwarew to each chassis, select it for use and if its not the presently running firmware, reboot the chassis

Additional options prompted for:

- parallel vs serial update
- selection of firmware files or directory containing .pkg files
- prompting for chassis password (default is to have password in fastfabric.conf)

If any chassis fails to be updated, use the `View ibtest result files` option to review the result files from the update. Refer to the section “Interpreting the ibtest log files” on page 5-68 for more details.

4.3.4
Show Status of Chassis IB Ports

(Switch) This will run the showallports -C command to allow the state and symbol error counts of all chassis ports to be manually reviewed.

(All) Instead it is recommended to run:

```
iba_report -i 10 -o errors -o slowlinks
```

on the IB Management node. This will check all the ports in the fabric for any links which have high error rates or are running at a lower speed than expected. Any such identified links should be diagnosed and corrected.

4.3.5
Reboot Chassis

(Switch) This will run the `ibtest -C` reboot command to reboot all the selected chassis and ensure they go down and come back up (as verified via ping over the management network).

4.3.6
Generate all Chassis Problem Report Information

(Switch) This will run the `captureall -C` command to collect configuration and status information from all chassis and generate a single `*.tgz` file that can be sent to the Support Representative.

4.3.7
Run a command on all chassis

(Switch) This will run the `cmdall -C` command. A Chassis CLI command may be specified to be executed against all selected chassis.

4.3.8
View ibtest results files

(All) This permits viewing of the `test.log` and `test.res` files which reflect the results from `ibtest` runs (such as for updating Chassis Firmware or rebooting all chassis per menu items above). The user is also given the option to remove these files after viewing them.

If not removed, subsequent runs of `ibtest` from within the current directory will continue to append to these files.

4.4 SilverStorm Externally Managed IB Switch Administration via Fast Fabric

This menu is focused on administration of SilverStorm 9024FC externally managed switches.

```
SilverStorm Technologies Inc. IB Switch Admin Menu (4.1.1.0.15)
Fast Fabric Externally Managed Switch List:
/etc/sysconfig/iba/ibnodes

0) Edit Config and Select/Edit Switch Files  [ Skip  ]
1) Verify Switch via Firmware dump           [ Skip  ]
2) Update Switch Firmware                    [ Skip  ]
3) Reboot Switch                             [ Skip  ]
4) View ibtest result files                  [ Skip  ]

P) Perform the selected actions
N) Select None

X) Return to Previous Menu (or ESC)
```

4.4.1 3.4.1Edit Config and Select/Edit Chassis Files

(Switch) This will permit the `ibnodes` and `fastfabric.conf` files to be edited. The `ibnodes` file selected and created via this menu should not list the Fast Fabric host itself. After editing the two files, an opportunity is given to edit them again or continue forward.

```
Selected Chassis File: /etc/sysconfig/iba/chassis
Do you want to edit/review/change the files? [y]:
```

The default will repeat the editing process, answer `n` to proceed to continue forward.

Refer to the section “Selection of Switches” on page 5-7 for more details about the format of the ibnodes file.

4.4.2 Verify Switch via Firmware Dump

(Switch) Use of this option is not recommended.

4.4.3 Update Switch Firmware

(Switch) This will run the `ibtest -n` upgrade command to permit the switch firmware version to be updated and the switch node name to be set.

NOTE: Consult the relevant switch firmware release notes to ensure any prerequisites for the upgrade to the new firmware level have been met prior to performing the upgrade via Fast Fabric.

Prompts will guide the user through options:

- select - push firmware to each switch and select it for use on next reboot
- run - push firmware to each switch, select it for use and reboot switches

Additional options prompted for:

- parallel vs serial update
- select of firmware files or directory containing .emfw files

If any switches fail to be updated, use the `View ibtest result files` option to review the result files from the update. Refer to the section “Interpreting the ibtest log files” on page 5-68 for more details.

4.4.4 Reboot Switch

(Switch) This will run the `ibtest -n` reboot command to reboot all the selected switches.

4.4.5 View ibtest result files

(All) This permits viewing of the `test.log` and `test.res` files that reflect the results from `ibtest` runs (such as those for updating Switch Firmware or rebooting all switches per menu items above). The user is also given the option to remove these files after viewing them.

If not removed, subsequent runs of `ibtest` from within the current directory will continue to append to these files.

Section 5

Detailed Descriptions of Command LineTools

Some of the commands are only applicable when Linux is being used. They will be marked with **(Linux)**. Similarly some of the commands are only applicable when QuickSlver Linux IB software is being used on the hosts. Those will be marked with **(Host)**. All commands which are applicable only when SilverStorm IB Switches or IB Chassis are being used will be marked with **(Switch)**. All remaining commands are generally applicable to all environments and will be marked with **(All)**.

NOTE: Some of the Linux commands may be applicable to other Unix-like operating systems if it is desired to enable use of non-IB specific Fast Fabric tools (such as `cmdall`) against the given hosts.

The Fast Fabric tools are installed in directories which are part of the standard Linux root PATH. Most of the tools are installed in `/sbin`.

5.1 Common Tool Options

There are some common options to the assorted command line tools. These options are applicable to most of the tools:

5.1.1 -?

Will display Usage information for any of the commands (as will any invalid option)

5.1.2 -p

Runs the operation/command in parallel. This means the operation is performed simultaneously on batches of 20 hosts. As such this option allows the overall time of an operation to be much lower. However, a side effect is that any output from the command will be bursty and intermingled. Therefore this option should be used for commands where there is no output or the output is of limited interest. For some commands (such as **`scpall`**), this will perform the operation in a quiet mode to limit output. If the user wants to change the number of parallel operations export **`TEST_MAX_PARALLEL=#`** where **`#`** is the new number (such as 30).

For more advanced operations (such as **`ibtest`**), parallel operation is the default mode.

Parallel operation can also be disabled by setting **`FF_MAX_PARALLEL`** to **`1`**.

5.1.3 -S

Prompt for password for admin on chassis. By default Fast Fabric operations against SilverStorm chassis (such as **cmdall**, **captureall**, **showallports**, and **ibtest**) obtain the chassis admin password from the **FF_CHASSIS_ADMIN_PASSWORD** environment variable which may be directly exported or part of **fastfabric.conf**. Alternatively the **-S** option may be used on these commands in which case the chassis admin password will be prompted for interactively. The password is prompted for once and the same password is then used to login to each chassis during the operation.

NOTE: Newer versions of SilverStorm chassis firmware permit ssh keys to be configured within the chassis for secure password-less login. In which case there is no need to configure a FF_CHASSIS_ADMIN_PASSWORD and FF_CHASSIS_LOGIN_METHOD can be ssh. Consult the *SilverStorm 9000 Users Guide* for more information.

5.1.4 -C

Specifies that the given operation should be performed against chassis. By default Fast Fabric operations are performed against hosts. However, selected Fast Fabric commands (such as **cmdall**, **pingall**, **captureall**, and **ibtest**) can also operate against SilverStorm internally managed IB chassis. When **-C** is specified, the operation will be performed against chassis instead of hosts (and the selection of chassis options discussed below will be used).

5.1.5 -n or -I

Specifies that the given operation should be performed against externally-managed switches (such as the SilverStorm 9024FC model IB switch). By default Fast Fabric operations are performed against hosts. However, selected Fast Fabric commands (such as **ibtest**) can also operate against externally-managed switches. When specified, the operation will be performed against switches instead of hosts (and the selection of switches options discussed below will be used).

NOTE: Some commands use **-n** while others use **-I**. In a future release this will be made consistent among all commands.

5.1.6
Selection of Hosts

For operations that are performed against a set of hosts, there are multiple ways to specify the hosts on which to operate:

1. Small sets of hosts can be easily specified on the command line via the **-h** option discussed below.
2. When multiple commands are performed against the same small set of hosts, the environment variable **HOSTS** can be used to specify a space separated lists of hosts.
3. For groups of hosts that will be used often, a file may be created listing the hosts. The defauilt file is **/etc/sysconfig/iba/hosts** that should list all hosts in the cluster except the host running Fast Fabric itself. Such a file may then be specified via the **-f** command line option or the **HOSTS_FILE** environment variable.

Within the tools the options are considered in the following order, the first item listed below that is specified is used for the given command.

1. **-h** option
2. **HOSTS** environment variable
3. **-f** option
4. **HOSTS_FILE** environment variable
5. **/etc/sysconfig/iba/hosts** file

For example if the **-h** option is used and the **HOSTS_FILE** environment variable is also exported, the command will operate only on hosts specified via the **-h** option.

5.1.6.1
Host List Files

The **-f** option or the **HOSTS_FILE** environment variable may be used to provide the name of a file containing the list of hosts on which to operate. The default is **/etc/sysconfig/iba/hosts**. In some fabrics it may be useful to create multiple files in **/etc/sysconfig/iba** representing different subsets of the fabric from which the user may operate. For example:

/etc/sysconfig/iba/hosts-mpi: list of MPI hosts

/etc/sysconfig/iba/hosts-fs: list of file server hosts

/etc/sysconfig/iba/hosts: list of all hosts except for the Fast Fabric node

/etc/sysconfig/iba/allhosts: list of all hosts including the Fast Fabric node

If a relative path is specified for the **-f** option or **HOSTS_FILE**, the current directory will be checked first, followed by **/etc/sysconfig/iba/**

5.1.6.1.1
Host List File Format

Below is a sample host list file:

```
# this is a comment
192.168.0.4# host identified by IP address
n001   # host identified by resolvable TCP/IP name
include /etc/sysconfig/iba/hosts-mpi # included file
```

Each line of the host list file may specify a single host, a comment or another host list file to include.

Hosts may be specified by IP address or a resolvable TCP/IP host name. Typically, host names are used for readability. Also, some Fast Fabric tools will translate the supplied host names to IPoIB hostnames, in which case names are generally easier to translate than numeric IP addresses. Typically management network hostnames are specified. However, if desired, IPoIB hostnames or IP addresses may be used. This can accelerate large file transfers and other operations.

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute pathnames. If relative pathnames are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line. By using a **#** to precede the comment. On lines with hosts or include directives, the **#** must be white space separated from any preceding hostname, IP address or included file name.

5.1.6.2
Explicit host names

When hosts are explicitly specified via the **-h** option or the **HOSTS** environment variable, a space separated list of host names (or IP addresses) may be supplied. For example: **-h 'host1 host2 host3'**.

5.1.7
Selection of Chassis

For operations which are performed against a set of chassis, there are multiple ways to specify the chassis on which to operate:

1. Small sets of chassis can .be easily specified on the command line via the **-H** option discussed below
2. When multiple commands will be performed against the same small set of chassis, the environment variable **CHASSIS** can be used to specify a space separated lists of chassis.
3. For groups of chassis which will be used often, a file may be created listing the chassis. The defauilt file is **/etc/sysconfig/iba/chassis** which should list all chassis in the cluster. Such a file may then be specified via the **-F** command line option or the **CHASSIS_FILE** environment variable.

Within the tools the options are considered in the following order, the first item listed below that is specified is used for the given command.

1. **-H** option
2. **CHASSIS** environment variable
3. **-F** option
4. **CHASSIS_FILE** environment variable
5. **/etc/sysconfig/iba/chassis** file

For example if the **-H** option is used and the **CHASSIS_FILE** environment variable is also exported, the command will operate only on chassis specified via the **-H** option.

5.1.7.1

Chassis List Files

The **-F** option or the **CHASSIS_FILE** environment variable may be used to provide the name of a file containing the list of SilverStorm IB chassis to operate on. The default is **/etc/sysconfig/iba/chassis**. In some fabrics it may be useful to create multiple files in **/etc/sysconfig/iba** representing different subsets of the fabric the user may operate from. For example:

/etc/sysconfig/iba/chassis-core: list of core switching chassis

/etc/sysconfig/iba/chassis-edge: list of edge switching chassis

/etc/sysconfig/iba/esm_chassis: list of chassis running an SM

/etc/sysconfig/iba/chassis: list of all chassis

If a relative path is specified for the **-F** option or **CHASSIS_FILE**, the current directory will be checked first, followed by **/etc/sysconfig/iba/**.

5.1.7.1.1
Chassis List File Format

Below is a sample chassis file:

```
# this is a comment
192.168.0.5# chassis IP address
edge1  # chassis resolvable TCP/IP name
include /etc/sysconfig/iba/corechassis # included file
```

Each line of the chassis list file may specify a single chassis, a comment or another chassis that list file to include.

A chassis may be specified by chassis management network IP address or a resolvable TCP/IP name. Typically names are used for readability.

Files to be included may be specified via an **`include`** directive followed by a file name. File names specified should generally be absolute path names. If relative path names are used, they will be searched for within the current directory then **`/etc/sysconfig/iba`**.

Comments may be placed on any line. By using a **`#`** to precede the comment. On lines with chassis or **`include`** directives, the **`#`** must be white space separated from any preceding name, IP address or included file name.

5.1.7.2
Explicit Chassis names

When chassis are explicitly specified via the **`-H`** option or the **`CHASSIS`** environment variable, a space separated list of names (or IP addresses) may be supplied. For example: **`-H 'chassis1 chassis2 chassis3'`**.

5.1.7.3
Selection of slots within a chassis

Normally, operations are performed against the management card in the chassis. For operations such as **`cmdall`**, the command is executed against the management interface for the given chassis. For more sophisticated operations such as firmware update, a directory with firmware for each chassis card type can be supplied and all cards in the chassis will be updated with the appropriate firmware from that directory.

However, in some cases it may be desirable to perform operations against a specific subset of cards within the chassis. In this case the chassis IP address or name within a chassis list or a chassis file can be augmented with a list of slot numbers on which to operate. This is done in the form:

```
chassis:slot1,slot2,…
```

For example:

```
i9k229:0
i9k229:0,1,5
192.168.0.5:0,1,5
```

NOTE: There must be no spaces within the chassis name and/or slot list.

This format is used by **cmdall** and **chassis firmware update**. This format may be used anyplace a chassis name or IP address is valid, such as the **-H** option, the **CHASSIS** environment variable or **chassis list files**. The slot number specified is ignored on some operations (such as **pingall**). Only slots containing management cards, EVICs and FVICs may be specified with this format. For all 9000 series chassis, **slot 0** is always an alias for the presently active management card for the chassis. For the remainder of slot usages in the chassis, the **chassisQuery** command can be executed against a given chassis to identify which slots have management, EVIC or FVIC cards.

NOTE: For any operation, care should be taken that a given chassis is listed only once with all relevant slots as part of that single specification. This is important so that parallel operations do not cause conflicting concurrent operations against a given chassis.

5.1.8 Selection of Switches

For operations that are performed against a set of fixed configuration externally-managed switches, there are multiple ways to specify the switch on which to operate:

1. Small sets of switches can be easily specified on the command line via the **-N** option discussed below.
2. When multiple commands will be performed against the same small set of switches, the environment variable **IBNODES** can be used to specify a space separate lists of switches
3. For groups of switches which will be used often, a file may be created listing the switches. The defauilt file is **/etc/sysconfig/iba/ibnodes** that should list all switches in the cluster. Such a file may then be specified via the **-L** command line option or the **IBNODES_FILE** environment variable.

Within the tools the options are considered in the following order, the first item listed below which is specified is used for the given command.

1. **-N** option
2. **IBNODES** environment variable
3. **-L** option

4. **IBNODES_FILE** environment variable
5. **/etc/sysconfig/iba/ibnodes** file

For example if the **-N** option is used and the **IBNODES_FILE** environment variable is also exported, the command will operate only on switches specified via the **-N** option.

5.1.8.1
Switch List Files

The **-L** option or the **IBNODES_FILE** environment variable may be used to provide the name of a file containing the list of SilverStorm IB switches on which to operate. The default is **/etc/sysconfig/iba/ibnodes**. In some fabrics it may be useful to create multiple files in **/etc/sysconfig/iba** representing different subsets of the fabric from which the user may operate.

If a relative path is specified for the **-L** option or **CHASSIS_FILE**, the current directory will be checked first, followed by **/etc/sysconfig/iba/**.

5.1.8.1.1
Switch List File Format

Below is a sample switch list file:

```
# this is a comment
0x00066a00d9000138,i9k138 # Node GUID with desired Name
0x00066a00d9000139,i9k139 # Node GUID with desired Name
include /etc/sysconfig/iba/moreswitches # included file
```

Each line of the switch list file may specify a single switch, a comment or another switch list file to include.

Switches can be specified by node GUID optionally followed by a comma and the IB Node Description (i.e., the name) to be assigned to the switch. The GUID will be used to select the switch and on firmware update operations, the node description will be written to the switch such that other Fast fabric tools (such as **saquery** and **iba_report**) can provide a more easily readable name for the switch.

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute path names. If relative path names are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line. By using a **#** to precede the comment. On lines with chassis or include directives, the **#** must be white space separated from any preceding GUID, name or included filename.

It is recommended that a unique node description be specified for each switch. This name should follow typical naming rules and use the characters a-z, A-Z, 0-9, and underscore. No spaces are allowed in the node description. Additionally, names should not start with a digit.

For 9024FC switches, the node GUID can be found on a label on the bottom of the switch. Alternately, the node GUIDs for switches in the fabric can be found use a command such as:

```
saquery -t sw -o nodeguid
```

NOTE: The above command will report all switch node GUIDs, including those of internally-managed chassis such as the 9120 model. GUIDs for internally-managed chassis cannot be specified for use in **-N**, **IBNODES**, **-L**, or **IBNODES_FILE** specified lists.

5.1.8.2
Explicit Switch names

When switches are explicitly specified via the **-N** option or the **IBNODES** environment variable, a space separated list of GUIDs (optionally with name) may be supplied. For example:

```
-N '0x00066a00d9000138,i9k138 0x00066a00d9000139,i9k139'
```

5.1.9
Selection of local Ports (subnets)

Some of the fabric health commands (**fabric_analysis**, **all_analysis**) permits a specific set of local HCA ports to be used for fabric analysis. The default is to use the first active port. However, for IB management nodes connected to more than 1 IB subnet, it is necessary to specify the local HCA and port such that the desired subnet will be analyzed. When the non-default behavior is desired, there are multiple ways to specify the local ports to use:

1. Small sets of ports can be easily specified on the command line via the **-p** option discussed below.
2. When multiple commands will be performed against the same small set of ports, the environment variable **PORTS** can be used to specify a space separated lists of ports.
3. For groups of ports that will be used often, a file may be created listing the ports. The defauilt file is **/etc/sysconfig/iba/ports** that should list all local ports connected to unique subnets. Such a file may then be specified via the **-t** command line option or the **PORTS_FILE** environment variable.

Within the tools the options that are considered in the following order, the first item listed below that is specified is used for the given command.

1. **-p** option
2. **PORTS** environment variable
3. **-t** option
4. **PORTS_FILE** environment variable
5. **/etc/sysconfig/iba/ports** file
6. default of the first active port on system (0 :0 port specification)

For example, if the **-p** option is used and the **PORTS_FILE** environment variable is also exported, the command will operate only on ports specified via the **-p** option.

5.1.9.1
Port List Files

The **-t** option or the **PORTS_FILE** environment variable may be used to provide the name of a file containing the list of local HCA ports to use. The default is **/etc/sysconfig/iba/ports**. In some fabrics it may be useful to create multiple files in **/etc/sysconfig/iba** representing different subsets of the ports from which the user may operate. For example:

/etc/sysconfig/iba/ports-primary: ports for which this node is primary

/etc/sysconfig/iba/ports-plain1: port(s) for plain1 subnet

/etc/sysconfig/iba/ports: list of all unique subnet ports

If a relative path is specified for the **-t** option or **PORTS_FILE**, the current directory will be checked first, followed by **/etc/sysconfig/iba/**.

5.1.9.1.1
Port List File Format

Below is a sample port list file:

```
# this is a comment
1:1 # first port on 1st HCA
1:2 # second port on 1st HCA
2:1 # first port on 2nd HCA
3:0 # first active port on 3rd HCA
include /etc/sysconfig/iba/ports-plain2# included file
```

Each line of the port list file may specify a single port, a comment or include another port list file.

Ports are specified as **hca:port**. No spaces are permitted. The first HCA is 1 and the first Port is 1. The value **0** for HCA or Port has special meaning. The allowed formats are:

```
0:0 = 1st active port in system
0:y = port y within system
x:0 = 1st active port on HCA x
x:y = HCA x, port y
```

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute pathnames. If relative pathnames are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line. By using a **#** to precede the comment. On lines with a port or **include** directive, the **#** must be white space separated from any preceding port or included filename.

5.1.9.2
Explicit ports

When ports are explicitly specified via the **-p** option or the **PORTS** environment variable, a space separated list of ports may be supplied.. For example: **-p '1:1 1:2 2:1'**.

5.2
Basic Setup and Administration Tools

5.2.1
pingall

(All): Pings a group of hosts or chassis to verify that they are powered on and accessible via TCP/IP ping

Usage:

```
pingall [-Cp] [-f hostfile] [-F chassisfile] [-h 'hosts'] [-H
'chassis']
```

-C - performs a ping against a chassis. The default is hosts

-p - ping all hosts/chassis in parallel

-f hostfile - file with hosts in cluster, default is **/etc/sysconfig/iba/hosts**

-F chassisfile - file with chassis in cluster default is **/etc/sysconfig/iba/chassis**

-h hosts - list of hosts to ping

-H chassis - list of chassis to ping

Example:

```
pingall
pingall -h 'arwen elrond'
HOSTS='arwen elrond' pingall
pingall -C
pingall -C -H 'chassis1 chassis2'
CHASSIS='chassis1 chassis2' pingall -C
```

Environment Variables:

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above

CHASSIS, CHASSIS_FILE - see discussion on selection of chassis above

FF_MAX_PARALLEL - when -p option is used maximum number of parallel operations to perform at once.

5.2.2 check_rsh

(Linux) Verifies that **rsh** is set up to allow passwordless file copies (RCP) and commands (**rsh**) to be run from this host to all the other hosts (and to itself via **localhost**) as a specific user (default is **root**). Additionally, this command can be used to verify **rsh** is setup to allow MPI to use **rsh** for job startup.

NOTE: For security reasons, configuration and use of **rsh/rcp/rlogin** is no longer recommended. Instead **ssh** is recommended. **SSH** may be used by MPI as well as **setup_ssh**.

Usage:

```
check_rsh [-i ipoib_suffix] [-f hostfile] [-h 'hosts'] [-u user]
```

-i 'ipoib_suffix '- suffix to apply to host names to create IPoIB host names. The default is **'-ib'**. Use **-i ''** to indicate no suffix.

-h hosts - list of hosts to setup

-f hostfile - file with hosts in cluster, default is **/etc/sysconfig/iba/hosts**

-u user - user on remote system to verify this user can rsh to. The default is current user code.

Example:

```
check_rsh
check_rsh -h 'arwen elrond'
HOSTS='arwen elrond' check_rsh
```

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above

5.2.3 setup_ssh

(Linux): creates ssh keys and configures them on all hosts so the system can ssh and scp into all other hosts without a password prompt. Typically, during cluster setup this tool is used to enable the root user on the IB Management node to login to the other hosts via password-less ssh. However, if desired, this tool can also aid the setup of password-less ssh login for other user codes as well.

Usage:

```
setup_ssh [-C] [-s] [-i ipoib_suffix] [-f hostfile] [-h 'hosts']
[-u user] [-S]
```

-C - only perform connect (to enter in local hosts knownhosts). When run in this mode, **-S** and **-s** options are ignored).

-s - use ssh/scp to transfer files, default is **rsh/rcp**.

-i ipoib_suffix - suffix to apply to host names to create IPoIB host names. The default is **'-ib'**.

-h hosts - list of hosts to setup.

-f hostfile - file with hosts in cluster, default is **/etc/sysconfig/iba/hosts**.

-u user - user on remote system to allow this user to ssh to, default is current user code.

-S - securely prompt for password for user on remote system.

Example:

```
setup_ssh -s -S -I""
setup_ssh -C
setup_ssh -h 'arwen elrond' -C
HOSTS='arwen elrond' setup_ssh -C
```

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above.

FF_IPOIB_SUFFIX - suffix to append to hostname to create IPoIB hostname. Used in absence of **-i**.

Fast Fabric provides additional flexibility in the translation between IPoIB and management network hostnames. Refer to appendix C for more information.

Setup_ssh provides an easy way to create ssh keys and distribute them to the hosts in the cluster. Many of the Fast Fabric tools (as well as many versions of MPI) require ssh be set up for password-less operation. Therefore, **setup_ssh** is an important setup step.

This tool also sets up ssh to the local host and the local hosts IPoIB name. This capability is required by selected Fast Fabric commands and may be used by some applications (such as MPI).

Setup_ssh has two modes of operation. The mode is selected by the presence or absence of the **-C** option. Typically, **setup_ssh** will first be run without the **-C** option, then it may later be run with the **-C** option.

Initial key exchange

When run without the **-C** option, **setup_ssh** will perform the initial key exchange and enable password-less ssh and scp. The key exchange can be accomplished using ssh and scp (in a password prompting manner) via the **-s** option or using password-less rsh and rcp (omitting the **-s** option).

The preferred way to use **setup_ssh** for initial key exchange is with the **-s** and **-S** options. This requires all hosts have been configured with the same password for the specified "user" (typically **root**). In this mode the password will be prompted for once and then ssh and scp are used in conjunction with that password to complete the setup for the hosts. Use in this manner also avoids the need to setup rsh/rcp/rlogin (which can be a security risk).

If **-s** is used without the **-S** option, the user will be prompted by ssh and scp for each host as they are setup. There will be multiple prompts per host. For a handful of hosts this is manageable, however for a significant number of hosts this can become cumbersome. Therefore, the **-S** option is recommended in this case.

If the **-s** option is not specified, rsh and rcp will be used to perform the ssh key exchange. This requires password-less rcp and rlogin be enabled on each host (**check_rsh** can perform verification).

Setup_ssh will configure password-less ssh/scp for both the management network and IPoIB. Typically, the management network will be used for Fast Fabric while IPoIB will be used for MPI and other applications. If IPoIB is not yet running (for example, during initial cluster installation IB software will not yet be installed on all the hosts), the **-i** option can be specified with an empty string:

```
setup_ssh -i ''
```

This will cause the last part of the setup of ssh for IPoIB to be skipped.

Refreshing local systems known hosts

If hosts have IP addresses added (for example by installing IB software and enabling IPoIB), IP addresses changes, MAC addresses changed or other aspects have changed (such as server OS reinstallation), the local hosts ssh known_hosts file can be refreshed by running **setup_ssh** with the **-C** option. This option will not transfer the keys, but rather will connect to each host (management network and IPoIB) in order to refresh the ssh keys. Existing entries for the specified hosts are replaced within the local **known_hosts** file. When run in this mode the **-S** and **-s** options are ignored. This mode assumes ssh has previously been setup for the hosts, as such no files are transferred to the specified hosts and no passwords should be required.

Typically after completing the installation and booting of IB software, **setup_ssh** will need to be rerun with the **-C** option to update the **knownhosts** file

5.2.4 cmdall

(Linux and Switch): Executes a command on all hosts or SilverStorm IB chassis. This is very powerful and can be used for everything from configuring servers or chassis, verifying that they are running, starting and stopping host processes, etc.

Usage:

```
cmdall [-Cpq] [-f hostfile] [-F chassisfile] [-h 'hosts'] [-H
'chassis'] [-u user] [-S] 'cmd'
```

-C - perform command against chassis, default is hosts

-p - run command in parallel on all hosts

-q - quiet mode, do not show command to execute

-f hostfile - file with hosts in cluster, default is **/etc/sysconfig/iba/hosts**

-F chassisfile - file with chassis in cluster default is **/etc/sysconfig/iba/chassis**

-h hosts - list of hosts on which to execute the command

-H chassis - list of chassis on which to execute the command

-u user - the user to perform the command as. For hosts, the default is current user code. For chassis, the default is **admin** (this argument is ignored)

-S - securely prompt for password for admin on chassis

Host Examples:

```
cmdall date
cmdall 'uname -a'
cmdall -h 'elrond arwen' date
HOSTS='elrond arwen' cmdall date
Chassis Examples:
cmdall -C 'ismPortStats'
cmdall -C -H 'chassis1 chassis2' ismPortStats
CHASSIS='chassis1 chassis2' cmdall ismPortStats
```

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above

CHASSIS, CHASSIS_FILE - see discussion on selection of chassis above

FF_MAX_PARALLEL - when -p option is used maximum number of parallel operations to perform at once.

FF_CHASSIS_LOGIN_METHOD - how to login to chassis. Can be ssh or telnet

FF_CHASSIS_ADMIN_PASSWORD - password for admin on all chassis. Used in absence of **-S** option.

NOTE: All commands performed with **cmdall** must be non-interactive in nature. **cmdall** will wait for the command to complete before proceeding. For example, when running host commands such as **rm**, the **-i** option (interactively prompt before removal) should not be used (Note that this option is sometimes part of a standard bash alias list). Similarly, when running chassis commands such as **fwUpdateChassis**, the **-reboot** option should not be used (this option causes an immediate reboot therefore, the command never returns). Similarly, the chassis command reboot should not be executed via **cmdall**. Instead use the **ibtest -C reboot** Fast Fabric command to reboot one or more chassis For further information about individual chassis CLI commands consult the *SilverStorm 9000 CLI Reference Guide*. For further information about Linux OS commands, consult the Linux man pages and any other documentation supplied with the OS by the OS supplier.

When performing **cmdall** against hosts, internally ssh is used. The command **cmdall** requires that password-less ssh be setup between the host running Fast Fabric and the hosts **cmdall** is operating against. The **setup_ssh** Fast Fabric tool can aid in setting up password-less ssh.

When performing **cmdall** against a set of chassis, all chassis must be configured with the same admin password.

For operations against chassis use of the **-S** option is recommended. This avoids the need to keep the password in configuration files.

5.2.5 captureall

(Switch and Host): Captures supporting information for a problem report from all hosts or SilverStorm IB chassis and uploads to this system

Usage:

```
captureall [-Cp] [-f hostfile] [-F chassisfile] [-h 'hosts']
[-H 'chassis'] [-d upload_dir] [-S] [file]
```

-C - perform capture against chassis, default is hosts

-p - perform capture in parallel

[for a host capture this only affects the upload phase]

-f hostfile - file with hosts in cluster, default is **/etc/sysconfig/iba/hosts**

-F chassisfile - file with chassis in cluster, default is **/etc/sysconfig/iba/chassis**

-h hosts -a list of hosts to perform a capture of

-H chassis - a list of chassis to perform a capture of

-d upload_dir - directory to upload to, default is uploads. If not specified, the environment variable **UPLOADS_DIR** will be used. If that is not exported, the default (**./uploads**) will be used.

-S - securely prompt for password for administrator on a chassis

file - name for capture file [**.tgz** will be appended]

When a host captureall is performed, **iba_capture** will be run to create the specified capture file within **~root** on each host (with the **.tgz** suffix added). The files will be uploaded and unpacked into a matching directory name within **upload_dir/hostname/** on the local system. The default file name is **hostcapture**.

When a chassis capture all is performed, the **chassis capture** CLI command will be run on each chassis and its output will be saved to **upload_dir/chassisname/file** on the local system. The default file name is **chassiscapture**.

For both host and chassis capture, the uploaded captures will be combined into a **tgz** file with the file name specified and the suffix **.all.tgz** added

Host Capture Examples:

captureall

The above example creates a hostcapture directory in **./uploads/<HOSTNAME>/** for each host in **/etc/sysconfig/iba/hosts** then creates **hostcapture.all.tgz**.

```
captureall mycapture
```

The above example creates a **mycapture** directory in **./uploads/<HOSTNAME>/** for each host in **/etc/sysconfig/iba/hosts** then creates **mycapture.all.tgz**.

```
captureall -h 'arwen elrond' 030127capture
```

Chassis Capture Examples:

```
captureall -C
```

The above example creates a **chassiscapture** file in **./uploads/<CHASSISNAME>/** for each chassis in **/etc/sysconfig/iba/chassis** then creates **chassiscapture.all.tgz**.

```
captureall -C mycapture
```

The above example creates a **mycapture.tgz** file in **./uploads/<CHASSISNAME>/** for each chassis in **/etc/sysconfig/iba/chassis** then creates **mycapture.all.tgz**.

```
captureall -C -H 'chassis1 chassis2' 030127capture
```

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above.

CHASSIS, CHASSIS_FILE - see discussion on selection of chassis above.

UPLOADS_DIR - directory to upload to, used in absence of **-d**.

FF_MAX_PARALLEL -maximum number of parallel operations to perform at once.

FF_CHASSIS_LOGIN_METHOD - how to login to chassis. Can be SSH or telnet.

FF_CHASSIS_ADMIN_PASSWORD - password for administrator on all chassis. Used in absence of **-S** option.

When performing captureall against hosts, internally SSH is used. The command **captureall** requires that password-less SSH be setup between the host running Fast Fabric and the hosts **captureall** is operating against. The **setup_ssh** Fast Fabric tool can aid in setting up password-less SSH.

When performing **captureall** against a set of chassis, all chassis must be configured with the same administrator password.

For operations against chassis use of the **-s** option is recommended. This avoids the need to keep the password in configuration files.

NOTE: The resulting host capture files can require significant amounts of space on the Fast Fabric host. Actual size will vary, but sizes can be multiple megabytes per host. As such it is recommended to ensure adequate space is available on the Fast Fabric system. In many cases it may not be necessary to run **captureall** against all hosts or chassis, but rather a representative subset may be sufficient. Consult with your support representative for further information.

5.3 File Management Tools

The following tools aid in copying files to and from large groups of nodes in the fabric.

Internally, these tools make use of SCP and require that password-less SSH/SCP be setup between the host running Fast Fabric and the hosts files that are being transferred to and from. The **setup_ssh** Fast Fabric tool can aid in setting up password-less SSH/SCP.

5.3.1 scpall

(Linux): The **scpall** tool permits efficient copying of files or directories from the current system to multiple hosts in the fabric. When copying large directory trees, performance can be improved by using the **-t** option. This will tar and compress the tree, then transfer the resulting compressed tarball to each node (and untar it on each node).

This can provide a powerful facility for copying data files, operating system files or even applications to all the hosts (or a subset of hosts) within the fabric.

Usage:

```
scpall [-p] [-r] [-f hostfile] [-h 'hosts'] [-u user] source_file
... dest_file

scpall -t [-p] [-f hostfile] [-h 'hosts'] [-u user] [source_dir
[dest_dir]]
```

Options are:

-r - recursive copy of directories

-p - perform copy in parallel

-t - optimized recursive copy of directories using tar

-h hosts - list of hosts to copy to

`-f hostfile` - file with hosts in cluster, default is **`/etc/sysconfig/iba/hosts`**.

`-u user` - user to perform copy to, default is current user code

`source_file`: the name of files to copy from this system, relative to the current directory. Multiple files may be listed.

`source_dir`: the name of directory to copy from this system, relative to the current directory.

`dest_file or dest_dir`: is the name of the file or directory on the destination system to copy to. It is relative to the home directory of the specified user code (an absolute path name may be specified if desired).

When performing directory copies using the **`-t`** option, the destination directory is optional. If not specified it defaults to the present directory name. If both the source and destination directory names are omitted, they both default to the current directory name.

Example:

```
# copy a single file
scpall MPI-PMB /root/MPI-PMB
# efficiently copy an entire directory tree
scpall -p -t /opt/iba/src/mpi_apps /opt/iba/src/mpi_apps
# copy a group of files
scpall a b c /root/tools/
# copy to an explitly specified set of hosts
scpall -h 'arwen elrond' a b c /root/tools
HOSTS='arwen elrond' scpall a b c /root/tools
```

NOTE: The tool **`scpall`** can only copy from this system to a group of systems in cluster. The **`user@`** style syntax cannot be used in the arguments to **`scpall`**.

Environment Variables

The following environment variables are also used by this command:

`HOSTS, HOSTS_FILE` - see discussion on selection of hosts above

`FF_MAX_PARALLEL` - when -p option is used maximum number of parallel operations to perform at once.

To copy from hosts in the cluster to this host, use **`uploadall`**.

5.3.2
uploadall

(Linux): Copies one or more files from a group of hosts to this system. Since the file name will be the same on each host, a separate directory on this system is created for each host and the file is copied to it. This is a convenient way to upload log files or configuration files for review. It can also be used in conjunction with **downloadall** to upload a host specific configuration file, edit it for each host and download the new version to all the hosts.

Usage:

```
uploadall [-rp] [-f hostfile] [-d upload_dir] [-h 'hosts'] [-u
user]  source_file ... dest_file
```

Options:

-p - perform copy in parallel on all hosts

-r - recursive upload of directories

-f <HOST FILE> - file with hosts in cluster, default is **/etc/sysconfig/iba/hosts**

-h hosts - list of hosts to upload from

-u user - user to perform copy to, default is current user code

-d upload_dir - directory to upload to, default is **uploads**. If not specified the environment variable **UPLOADS_DIR** will be used, if that is not exported the default (`./uploads`) will be used.

source_file - the name of files to copy to this system, relative to the current directory. Multiple files may be listed.

dest_file - is the name of the file or directory on this system to copy to. It is relative to **upload_dir/<HOSTNAME>**.

A local directory within **upload_dir/** will be created for each host being uploaded from. Each uploaded file will be copied to **upload_dir/<HOSTNAME>/dest_file**. If more than one source file is specified, **dest_file** will be treated as a directory name and the directories **upload_dir/<HOSTNAME>/dest_file/** will be created for each host and the **source_files** will be uploaded to those directories.

Example:

```
# upload two files from 2 hosts
uploadall -h 'arwen elrond' capture.tgz /etc/init.d/ipoib.cfg .
# upload two files from all hosts
uploadall capture.tgz /etc/init.d/ipoib.cfg .
# upload network config files from all hosts
uploadall -r -p /etc/sysconfig/network-scripts network-scripts
# upload two files to a specific subdirectory of upload_dir
uploadall capture.tgz /etc/init.d/ipoib.cfg pre-install
```

The above example copies **capture.tgz** and **/etc/init.d/ipoip.cfg** to **./uploads/<HOSTNAME>/preinstall/** where there a **<HOSTNAME>** directory is created for each host in **/etc/sysconfig/iba/hosts**.

NOTE: The **uploadall** tool can only copy from a group of systems in a cluster to this system. The **user@** style syntax cannot be used in the arguments to **uploadall**.

To copy files from this host to hosts in the cluster use **scpall** or **downloadall**.

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above.

FF_MAX_PARALLEL - when **-p** option is used maximum number of parallel operations to perform at once.

UPLOADS_DIR - the directory to upload to, used in absence of **-d**.

5.3.3 downloadall

(Linux): Copies one of more files to a group of hosts from a system. Since the file contents to copy may be different for each host, a separate directory on this system is used for the source files for each host. This can also be used in conjunction with **uploadall** to upload a host-specific configuration file, edit it for each host and download the new version to all the hosts.

Usage:

```
downloadall [-rp] [-f hostfile] [-d download_dir] [-h 'hosts']
[-u user] source_file ... dest_file
```

-p - perform copy in parallel on all hosts

-r - recursive download of directories

-f hostfile - file with hosts in cluster. The default is **/etc/sysconfig/iba/hosts**.

-h hosts - the list of hosts to download files to

-u user - the user to perform the copy. The default is current user code

-d download_dir - the directory to download files to. The default is **./downloads**. If not specified, the environment variable **DOWNLOADS_DIR** will be used. If that is not exported the default (**./downloads**) will be used.

source_file - the name of files to copy from the system. Multiple files may be listed. The option **source_file** is relative to **download_dir/<HOSTNAME>**.

A local directory within **download_dir/** must exist for each host being downloaded to Each downloaded file will be copied from **download_dir/<HOSTNAME>/source_file**.

dest_file - is the name of the file or directory on the destination hosts to copy to.

If more than one source file is specified, **dest_file** will be treated as a directory name. The given directory must already exist on the destination hosts (the copy will fail for hosts where the directory does not exist).

Example:

```
   # copy two files to 2 hosts
downloadall -h 'arwen elrond' ics_srp.cfg ics_inic.cfg
/etc/sysconfig
# copy two files to all hosts
   downloadall ics_srp.cfg ics_inic.cfg /etc/sysconfig
```

NOTE: The tool **downloadall** can only copy from this system to a group of hosts in the cluster. The **user@** style syntax cannot be used in the arguments to **downloadall**.

To copy files from hosts in the cluster to this host use **uploadall**.

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above.

FF_MAX_PARALLEL - when **-p** option is used maximum number of parallel operations to perform at once.

DOWNLOADS_DIR - directory to download from, used in absence of **-d**.

5.3.4
Simplified Editing of Node-Specific Files

(Linux): The combination of **`uploadall`** and **`downloadall`** provide a powerful yet simple to use mechanism for reviewing and/or editing node-specific files without the need to login to each node.

This is best explained with an example.

Assume the file **`/etc./sysconfig/network-scripts/ifcfg-ib1`** needs to be reviewed and possibly edited for each host. This file would typically contain the IP configuration information for IPoIB and may contain a unique IP address per host.

To upload the file from all the hosts:

```
uploadall /etc/sysconfig/network-scripts/ifcfg-ib1 ifcfg-ib1
```

Now edit the uploaded files with an editor, such as **`vi`**:

```
vi uploads/*/ifcfg-ib1
```

If via the editor the file was changed for some or all of the hosts, it can then be downloaded to all the hosts:

```
downloadall -d uploads ifcfg-ib1 /etc/sysconfig/network-scripts/ifcfg-ib1
```

Alternatively, if there was no need to download the file to all hosts, a subset of hosts can be specified using the **`-h`** option or by creating an alternate host list file:

```
downloadall -d uploads -h 'host1 host32' ifcfg-ib1 /etc/sysconfig/network-scripts/ifcfg-ib1
```

NOTE: When downloading to a subset of hosts, make sure that only the hosts uploaded from are specified.

5.3.5
Simplified Setup of Node-Generic Files

(Linux): In contrast **`scpall`** can provide a powerful yet simple to use mechanism for transferring files to all nodes that are generic (i.e., not node-specific).

For example, if all nodes in the cluster will use the same DNS server and TCP/IP name resolution, they may be quickly set as follows:

Create an appropriate local file with the desired information. For example:

```
vi resolv.conf
```

Now copy the file to all hosts:

```
scpall resolv.conf /etc/resolv.conf
```

5.4 Fabric Analysis Tools

5.4.1 Fabric_info

`Fabric_info` provides a brief summary of the components in the fabric. **`Fabric_info`** uses the first active IB port on the given local host to perform its analysis.

Example output:

```
Fabric_info
```

`Fabric_info` has no options and uses no environment variables.

Example output:

```
# fabric_info
Fabric Information:
SM: i9k229 Guid: 0x00066a00d8000229 State: Master
SM: i9k3ff Guid: 0x00066a00d90003ff State: Standby
Number of CAs: 17
Number of CA Ports: 22
Number of Switch Chips: 6
Number of Links: 29
Number of 1x Ports: 2
```

The output is as follows:

`SM` - each subnet manger (SM) running in the fabric is listed along with its node name, port GUID and present SM state (Master, Standby, etc).

`Number of CA` - number of unique channel adapters (CA) in the fabric. A CA with two-connected ports is counted as a single CA.

NOTE: Channel adapters include both HCAs in servers as well has TCAs within IO Modules, IB Native Storage, etc.

`Number of CA ports` - number of connected CA ports in the fabric.

`Number of Switch chips` - number of unique switches in the fabric.

NOTE: A large IB switch may be composed of many unique switch chips.

`Number of Links` - number of IB links in the fabric. Note that a large IB switch may have internal links.

Number of 1x Ports - number of ports in the fabric running at 1x speed. Typically such ports represent a bad cable connection, a bad cable, too long a cable or perhaps faulty hardware on one side of the link.

Fabric_info can be very useful as a quick assessment of the fabric state. **Fabric_info** can be run against a known good fabric to identify its components and then later run to see if anything has changed about the fabric configuration or state. When used in this manner it can be used to quickly identify if CAs are down, links are missing, SMs are missing, etc.

For more extensive fabric analysis, see **iba_report**.

5.4.2 showallports

(Switch and Host): Displays basic port state and statistics for all host nodes, chassis or externally managed switches.

NOTE: **iba_report** is a newer and more powerful Fast Fabric command. For general fabric analysis, use **iba_report** with options such as **-o errors** and/or **-o slowlinks** to perform a more efficient analysis of link speeds and errors.

Usage:

```
showallports [-C|-I] [-f hostfile] [-F chassisfile] [-L
ibnodefile] [-h 'hosts'] [-H 'chassis'] [-N 'ibnodes'] [-M
'host'] [-S]
```

-C - perform operation against chassis; the default is **hosts**

-I - perform operation against IB nodes; the default is **hosts**

-f hostfile - file with hosts in cluster; the default is **/etc/sysconfig/iba/hosts**

-F chassisfile - file with chassis in cluster; the default is **/etc/sysconfig/iba/chassis**

-L ibnodefile - a file with ib nodes in the cluster; the default is **/etc/sysconfig/iba/ibnodes**

-h hosts - a list of hosts to show port information

-H chassis - a list of chassis to show port information

-N ibnodes - a list of IB nodes to show port information

-M host - The management host. This is the remote host from which to run IB node queries; the default is **localhost**

-S - securely prompt for password for administrator on chassis

Environment:

HOSTS - a list of hosts, used if the **-h** option is not supplied

CHASSIS - a list of chassis, used if the **-C** is used and the **-h** option is not supplied

HOSTS_FILE - a file containing the list of hosts, used in absence of **-f** and **-h**

CHASSIS_FILE - a file containing the list of chassis, used in absence of **-F** and **-H**

Example:

```
showallports
showallports -h 'elrond arwen'
HOSTS-'elrond arwen' showallports
showallports -C
showallports -H 'chassis1 chassis2'
CHASSIS-'chassis1 chasssi2' showallports
    showallports -I
    showallports -I -N '0x00066a0005000105 0x00066a0005000110'
IBNODES='0x00066a0005000105 0x00066a0005000110' showallports
-I
```

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE - see discussion on selection of hosts above

CHASSIS, CHASSIS_FILE - see discussion on selection of chassis above

IBNODES, IBNODES_FILE - see discussion on selection of switches above

MGMT_HOST - host to use to perform IB node queries, used in absence of **-M**

FF_MAX_PARALLEL - when **-p** option is used maximum number of parallel operations to perform at once.

FF_CHASSIS_LOGIN_METHOD - how to login to chassis. Can be SSH or Telnet

FF_CHASSIS_ADMIN_PASSWORD - password for the administrator on all chassis. Used in absence of **-S** option.

When performing **showallports** against hosts, internally SSH is used. **showallports** requires that password-less SSH be setup between the host running Fast Fabric and the hosts **showallports** is operating against. The **setup_ssh** Fast Fabric tool can aid in setting up password-less SSH.

When performing **showallports** against a set of chassis, all chassis must be configured with the same administrator password.

For operations against chassis use of the **-S** option is recommended. This avoids the need to keep the password in configuration files.

When performing **showallports** against externally-managed switches it requires an IB-enabled management node with Fast Fabric installed. Typically this will be the Fast Fabric node from which **showallports** is being run. However, if desired an alternate node may be specified by the **-M** option or **MGMT_HOST** environment variable.

5.4.3 iba_report

(All): **iba_report** provides powerful fabric analysis and reporting capabilities. It must be run on a host connected to the IB fabric with Fast Fabric installed.

iba_report obtains all its data in an IBTA-compliant manner. Therefore, it will interoperate with both SilverStorm and 3rd party IB components, provided those components are IBTA compliant and implement the IBTA optional features required by **iba_report**.

iba_report requires that the subnet manager implement all the IBTA SA queries defined in the standard (such as SM Info records, Link Records, Trace Routes, Port Records, Node Records, etc). As such, it is recommended that the QuickSilver Fabric Manager version 4.0 or later be used. **iba_report** requires all end nodes to implement the PMA PortCounters (IBTA mandatory counters). Also any end nodes which report support of a IBTA device management agent must implement the IOU Info, IOC Profile and Service Entry queries as outlined in the IBTA 1.1 standard.

iba_report takes advantage of these interfaces to obtain extensive information about the fabric from the subnet manager and the end nodes. Using this information, **iba_report** is able to cross reference it and produce analysis greatly beyond what any single subnet manager request could provide. As such, it exceeds the capabilities previously available in tools such as **saquery** and **fabric_info**.

iba_report internally cross references all this information so its output can be in user-friendly form. Reports will include both GUIDs, LIDs and names for components. Obviously, these reports will be easiest to read if the end user has taken the time to provide unique names for all the components in the fabric (node names and IOC names). All SilverStorm components support this capability. For hosts, the node names automatically are assigned based on the network host name of the server. For switches and line cards the names can be assigned via the element managers for each component.

Each run of **iba_report** obtains up to date information from the fabric. At the start of the run **iba_report** will take a few seconds to obtain all the fabric data, then it will output it to **stdout**. The reports are sorted by GUIDs and other permanent information such that they can be rerun in the future and produce

output in the same order even if components have been rebooted. This is useful for comparison using simple tools like **diff**. **iba_report** permits multiple reports to be requested for a single run (i.e., 1 of each report type).

By default **iba_report** uses the first active port on the local system. However, if the IB management node is connected to more than one fabric (e.g., a subnet), the HCA and port may be specified to select the fabric to analyze.

Usage:

```
iba_report [-v][-q] [-h hca] [-p port] [-o report] [-d detail]
[-P|-H] [-N] [-x] [-s] [-i seconds] [-C] [-a] [-c file] [-L]
[-F point] [-S point] [-D point] [-Q]
```

-v/--verbose - verbose output

-q/--quiet - disable progress reports

-h/--hca hca - HCA to send via, default is 1st HCA

-p/--port port - port to send via, default is 1st active port

-o/--output report - report type for output

-d/--detail level - level of detail **0-n** for output, default is 2

-P/--persist - only include data persistent across reboots

-H/--hard - only include permanent hardware data

-N/--noname - omit node and IOC names

-x/--xml - output in XML

-s/--stats - get performance statistics for all ports

-i/--interval seconds - obtain performance statistics over interval seconds, clears all statistics, waits interval seconds, then generates report. Implies **-s**

-C/--clear - clear performance stats for all ports. Only stats with error thresholds are cleared. A clear occurs after generating the report.

-a/--clearall - clear all performance stats for all ports

-c/--config file - error thresholds configuration file. The default is **/etc/sysconfig/iba/iba_mon.conf**

-L/--limit - For port error counters check (**-o errors**) and port counters clear (**-C** or **-i**) with **-F** limit operation to exact specified focus. Normally the neighbor of each selected port would also be checked/cleared does not affect other reports

-F/--focus point - focus area for report used for all reports except route to limit scope of report

-S/--src point - source for trace route, default is local port

-D/--dest point - destination for trace route

-Q/--quietfocus - do not include focus description in report

Report Types:

comps - summary of all systems and SMs in fabric

brcomps - brief summary of all systems and SMs in fabric

nodes - summary of all node types and SMs in fabric

brnodes - brief summary of all node types and SMs in fabric

ious - summary of all IO units in the fabric

links - summary of all links

extlinks - summary of links external to systems

slowlinks - summary of links running slower than expected

slowconfiglinks- summary of links configured to run slower than supported

includes **slowlinks**

slowconnlinks- summary of links connected with mismatched speed potential

includes **slowconfiglinks**

misconfiglinks - summary of links configured to run slower than supported

misconnlinks - summary of links connected with mismatched speed potential

errors - summary of links whose errors exceed counts in the configuration file

otherports - summary of ports not connected to the fabric

all - comp, nodes, ious, links, extlinks, slowconnlinks, and errors reports

route - trace route between **-S** and **-D** points

none - no report, useful if just want to clear statistics

Point Syntax:

gid:value - value is numeric port gid of form: **subnet:guid**

lid:value - value is numeric lid

portguid:value - value is numeric port GUID

nodeguid:value - value is numeric node GUID

nodeguid:value1:port:value2 - value1 is numeric node GUID, value2 is port #

iocguid:value - value is numeric IOC GUID

iocguid:value1:port:value2 - value1 is numeric IOC GUID, value2 is port #

systemguid:value - value is numeric system image GUID

systemguid:value1:port:value2 - value1 is numeric system image GUID

value2 is port #

ioc:value - value is IOC Profile ID String (IOC Name)

ioc:value1:port:value2 - value1 is IOC Profile ID String (IOC Name)

value2 is port #

iocpat:value - value is global pattern for IOC Profile ID String (IOC Name)

iocpat:value1:port:value2 - value1 is global pattern for IOC Profile ID String

(IOC Name), value2 is port #

ioctype:value - value is IOC type (VNIC or SRP)

ioctype:value1:port:value2 - value1 is IOC type (VNIC or SRP)

value2 is port #

node:value - value is node description (node name)

node:value1:port:value2 - value1 is node description (node name)

value2 is port #

nodepat:value - value is glob pattern for node description (node name)

nodepat:value1:port:value2 - value1 is glob pattern for node description

(node name), value2 is port #

nodetype:value - value is node type (SW, CA or RT)

nodetype:value1:port:value2 - value1 is node type (SW, CA or RT)

value2 is port #

sm - master subnet manager

route:point1:point2 - all ports along the routes between the 2 given points

Examples:

`iba_report` can generate hundreds of different reports. Following is a list of some commonly generated reports:

Analyze a fabric for bad cables:

```
iba_report -o slowlinks -o errors
```

Analyze a fabric for bad cables or misconfigured ports:

```
iba_report -o slowconfiglinks -o errors
```

Analyze a fabric for bad cables or misconfigured ports or misconnected ports:

```
iba_report -o slowconnnlinks -o errors
```

Reverse lookup a lid:

```
iba_report -o brnodes -F lid:5
```

Reverse lookup a nodeguid:

```
iba_report -o brnodes -F nodeguid: 0x00066a0098000380
```

Reverse lookup a portguid:

```
iba_report -o brnodes -F portguid: 0x00066a00a0000380
```

Find all the connections to a server:

```
iba_report -o links -F node:duster
```

Find all the connections to a switch chip:

```
iba_report -o links -F 'node:i9k156'
```

Find all the connections to a multi-node system:

```
iba_report -o links -F systemguid:0x00066a0098000380
```

Report on all the components in a multi-node system:

```
iba_report -o comp -F node:goblin
```

Identify the routes between 2 servers:

```
iba_report -o route -S node:duster -D node:goblin
```

Identify the route between a server and a specific lid:

```
iba_report -o route -S node:duster -D lid:5
```

Identify the route between a server and the master SM:

```
iba_report -o route -S node:duster -D sm
```

Analyze the route between 2 nodes for bad cables or misconfigured ports or misconnected ports:

```
iba_report -o slowconnnlinks -o errors -F
route:node:cuda:node:duster
```

Identify the routes between this server and another server:

```
iba_report -o route -D node:goblin
```

Analyze a single switch for any high error counts:

```
iba_report -o errors -F 'node:i9k156'
```

Identify the routes between a server and an IOC:

```
iba_report -o route  -S node:duster -D 'ioc:Chassis
0x00066A005000010C, Slot 2, IOC 2'
```

Clear all the port counters in the fabric:

```
iba_report -C -o none
```

Clear all the port counters on a multi-HCA server:

```
iba_report -C -F node:goblin -o none
```

Check all port counters, clear them, then recheck:

```
iba_report -o errors -C; sleep 10; iba_report -o errors
```

Check all port counters, clear them, then recheck:

```
iba_report -o errors -C; sleep 10; iba_report -o errors
```

Clear all port counters, wait 10 seconds, then check

```
Iba_report -i 10 -o errors
```

Check all port counters on a server:

```
iba_report -o errors -F node:goblin
```

Check all port counters on a specific port on a server:

```
iba_report -o errors -F node:goblin:port:2
```

Get all the detailed information for a server including port counters:

```
iba_report -o nodes -F node:goblin -d 5 -s
```

Get all the detailed information for an IOU including port counters:

```
iba_report -o nodes -F 'ioc:Chassis 0x00066A005000010C, Slot
2, IOC 2' -d 5 -s
```

5.4.3.1

Basics of Using iba_report

`iba_report` can be run with no options at all. In this mode it provides a brief list of the nodes in the fabric (the brnodes report). The report organizes nodes as CAs, Switches and Routers. It also includes a summary of all the SMs in the fabric.

Here is a sample of iba_report for a small fabric:

```
[root@duster root]# iba_report
Node Type Brief Summary
14 Connected CAs in Fabric:
NodeGUID          Type Name
    Port LID   PortGUID           Width Speed
0x0002c9020020e0d4 CA coyote1
      1 0x000d 0x0002c9020020e0d5   4x   2.5Gb
0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot 2
      2 0x0014 0x00066a02580001e0   4x   2.5Gb
0x00066a0098000001 CA julio
      1 0x000c 0x00066a00a0000001   4x   2.5Gb
0x00066a00980001b8 CA orc
      1 0x000b 0x00066a00a00001b8   4x   2.5Gb
0x00066a0098000380 CA goblin
      1 0x000a 0x00066a00a0000380   4x   2.5Gb
0x00066a0098000384 CA cuda
      1 0x0005 0x00066a00a0000384   1x   2.5Gb
      2 0x0006 0x00066a01a0000384   4x   2.5Gb
0x00066a00980003a6 CA erik
      1 0x0015 0x00066a00a00003a6   4x   2.5Gb
      2 0x0016 0x00066a01a00003a6   4x   2.5Gb
0x00066a00980006a2 CA goblin
      1 0x000f 0x00066a00a00006a2   4x   2.5Gb
0x00066a0098000849 CA rockaway
      2 0x000e 0x00066a01a0000849   4x   2.5Gb
0x00066a0098002813 CA brady
      1 0x0002 0x00066a00a0002813   4x   2.5Gb
      2 0x0003 0x00066a01a0002813   4x   2.5Gb
0x00066a0098002854 CA brady
      1 0x0004 0x00066a00a0002854   4x   2.5Gb
      2 0x0008 0x00066a01a0002854   4x   2.5Gb
0x00066a0098003f81 CA ibm345
      1 0x0007 0x00066a00a0003f81   4x   2.5Gb
0x00066a009800447b CA duster
```

```
      1 0x0011 0x00066a00a000447b   4x   2.5Gb
      2 0x0012 0x00066a01a000447b   4x   2.5Gb
0x00066a0098004a73 CA erik
      1 0x0009 0x00066a00a0004a73   4x   2.5Gb

3 Connected Switches in Fabric:
NodeGUID          Type Name
    Port LID   PortGUID           Width Speed
0x00066a00280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev
A)
      0 0x0013 0x00066a00280002cd Noop    Noop
      3                             4x   2.5Gb
      5                             4x   2.5Gb
0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
      0 0x0001 0x00066a00d8000123   4x   2.5Gb
      1                             4x   2.5Gb
      2                             1x   2.5Gb
      3                             4x   2.5Gb
      4                             4x   2.5Gb
      5                             4x   2.5Gb
      6                             4x   2.5Gb
      7                             4x   2.5Gb
      8                             4x   2.5Gb
      9                             4x   2.5Gb
     10                             4x   2.5Gb
     11                             4x   2.5Gb
     12                             4x   2.5Gb
     14                             4x   2.5Gb
     15                             4x   2.5Gb
     16                             4x   2.5Gb
     17                             4x   2.5Gb
     18                             4x   2.5Gb
     19                             4x   2.5Gb
     20                             4x   2.5Gb
0x00066a10280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev
```

```
B)
      0 0x0010 0x00066a10280002cd Noop    Noop
      2                                4x   2.5Gb
      4                                4x   2.5Gb

1 Connected SMs in Fabric:
State       GUID               Name
Master      0x00066a00d8000123 InfiniCon Systems InfinIO9024
```

Each **iba_report** allows for various levels of detail. Increasing detail is shown as further indentation of the additional information. The **-d** option to **iba_report** controls the detail level. The default is **2**. Values from **0-n** are

permitted. The maximum detail per report varies, but most have less than 5 detail levels.

```
For example, the above report when run at detail level 0 outputs:
[root@duster root]# iba_report -d 0
Node Type Brief Summary
14 Connected CAs in Fabric:
3 Connected Switches in Fabric:
1 Connected SMs in Fabric:

You will notice this is a nice summary of fabric components and
is very similar to fabric_info.
At the next level of detail you get a report with a little more
detail:
[root@duster root]# iba_report -d 1
Node Type Brief Summary
14 Connected CAs in Fabric:
NodeGUID          Type Name
0x0002c9020020e0d4 CA coyote1
0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot 2
0x00066a0098000001 CA julio
0x00066a00980001b8 CA orc
0x00066a0098000380 CA goblin
0x00066a0098000384 CA cuda
0x00066a00980003a6 CA erik
0x00066a00980006a2 CA goblin
0x00066a0098000849 CA rockaway
0x00066a0098002813 CA brady
0x00066a0098002854 CA brady
0x00066a0098003f81 CA ibm345
0x00066a009800447b CA duster
0x00066a0098004a73 CA erik

3 Connected Switches in Fabric:
NodeGUID          Type Name
0x00066a00280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev
A)
```

```
0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
0x00066a10280002cd SW InfiniCon Systems InfiniFabric (Sw A Dev
B)


1 Connected SMs in Fabric:
State       GUID               Name
Master      0x00066a00d8000123 InfiniCon Systems InfinIO9024
```

The above examples were all performed with a single report, the brnodes (Brief Nodes) report. However this is just one of the many topology reports which **iba_report** can generate the others include:

- **nodes** - a more verbose form of brnode which can provide much greater levels of detail to drill down into all the details of every node, even down to all the port state, IOUs/IOCs/Services, Port counters.
- **comps** and **brcomps** are very similar to brnodes and nodes, except the reports are organized around systems. The grouping into systems is based on system image guids for each node. This report will help to present more complex systems (such as servers with multiple HCAs or large switches composed of multiple IB Switch chips).

NOTE: All SilverStorm switches implement a system image GUID and will therefore be properly grouped. However, some third-party devices do not implement the system image GUID and may report a value of 0. In such a case **iba_report** will treat each component as an independent system.

- **links** - This report presents all the links in the fabric. The output is very concise and helps to identify the connectivity between nodes in the fabric.
- **extlinks** - All the external links in the fabric (eg. those between different systems).
- **ious** - This is somewhat similar to the nodes reports, however the focus is around IOUs/IOCs and IO Services in the fabric. This report can be used to identify various IO devices in the fabric and their capabilities (such as the SilverStorm EVIC and FVIC Virtual IO Controllers or IBTA compliant direct-attach IB storage).
- **otherports** - All the ports which are not connected to this fabric. This report will identify additional ports on CAs or Switches which are not connected to this fabric. For switches these represent unused ports. For CAs these may be ports connected to other fabrics or unused ports.

The above reports are all summaries of the present state of the fabric. These reports can be very helpful to analyze the configuration of the fabric and or verify it was installed consistent with the desired design and configuration.

However, **iba_report** does not stop there. Additionally, **iba_report** has reports that will help to analyze the operational characteristics of the fabric and help to identify bottlenecks and faulty components in the fabric.

To assist in this area, iba_report also supports the following reports:

- **slowlinks** - identifies links which are running slower than expected. This helps to pinpoint bad cables or components in the fabric, such as a 4x cable that is poorly-connected and therefore only runs at 1x link width. The analysis includes both link speed and width.
- **slowconfiglinks** - this extends on the slowlinks report to also report links which have been configured (most likely by software) to run at a width or speed below their potential. Such as DDR capable links which have been forced to run at SDR rates.
- **slowconnlinks** - this further extends on the slowconfiglinks report to also report links which are cabled such that one of the ends of the link will never run to its potential. Such as a DDR capable HCA connected to an SDR switch.
- **misconfiglinks** - this is similar to **slowconfiglinks** in that it reports links which have been configured to run below their potential. However it does not include links which are running slower than expected.
- **misconnlinks** - this is similar to **slowconnlinks** in that it reports links which have been connected between ports of different speed potential. However it does not include links which are running slower than expected, nor links which have been configured to run slower than their potential.
- **errors** - this performs a single point in time analysis of the PMA port counters for every node and port in the fabric. All the counters are compared against configured thresholds (defaults are those in the **iba_mon.conf** file). Any link whose counters exceed these thresholds are listed (and depending on the detail level the exact counter and threshold will be reported). This is a powerful way to identify marginal links in the fabric such as bad or loose cables or damaged components.
- **route** - This permits the user to identify two end points in the fabric (by node name, node GUID, port name, port GUID, system image GUID, LID, port GID, IOC GIUD or IOC name) and obtain a list of all the links and components used when these two end points communicate. If there are multiple paths between the end points (such as a CA with 2 connected ports or a system with 2 CAs), the route for every available path (based on presently configured routing tables) will be reported.

The above set of reports can therefore be very powerful ways to obtain point in time status and problem analysis for the fabric.

5.4.3.2
Topology Verification

iba_report provides a flexible way to identify changes to the fabric or the appropriate reassembly of the fabric after a move (for example after staging and testing the fabric in a remote location before final installation at a customer site).

In this mode of operation, all the above reports are available, however the types of information output can be filtered. For example, using the **-P** option, information which would not persist across a fabric reboot (such as LIDs and error counters) will be omitted from the report (and marked out with **xxx**). Such a report can be saved for later comparison to a future report. Since **iba_report** produces simple text reports, standard tools such as **sdiff** (i.e., side by side diff) can be used for easy comparison and analysis of what changed.

Given the wealth of reports available, the user can select the information they want to save. For ease of use an **all** report is available which includes all the reports of general interest.

If software configuration changes are anticipated (such as adjusting the timeouts the SM configures in the fabric), the **iba_report -H** option can be used. This will further limit the report to only include hardware information. This is a superset of **-P** and omits more information.

A related but independent option is **-N**. This will omit all the node and IOC names from the report. If changes are anticipated in this area, this option can be used so future diffs will not report changes in names.

5.4.3.3
Focused Reports

One of the more powerful features of iba_report is the ability to focus a report on a subset of the fabric. Using the **-F** option the user can specify a node name, node name pattern, node guid, node type, port guid, IOC name, IOC name pattern, ioc guid, ioc type, system image guid, port gid, port rate, lid or SM. The subsequent report will indicate the total components in the fabric but will only report on those which relate to the focus area. For example in a nodes report, if a port is specified for focus, only the node containing that port will be reported on. In a links report, only the link using that port will be reported.

Notice that a focus level that is different from the orientation of the report may be chosen. For example if a node name is specified as the focus for the links report, a report of all the links to that node will be provided. This could include multiple switch ports or CA ports.

By carefully using this feature of report focus, reverse lookups can be done. For example, doing a **brnodes** report with a focus on a LID will reverse lookup the LID and indicate what node it is for.

When focusing a report, it can sometimes be helpful to also use a detail level of `0` or `1`. In this case the report will show only a count of number of matches (for detail `0`) and just the highest level of the entity which matches (for detail `1`).

5.4.3.3.1
Advanced Focus

The node name, node name pattern, node guid, node type, IOC name, IOC name pattern, IOC GUID, IOC type and system image GUID also allow for a port number specifier. This permits the focus to be limited to the given port number. If the selection resolves to multiple switches or CAs (for example a system composed of multiple nodes), all ports on the present fabric matching the given port number will be selected.

An even more advanced form of focus is to focus on the route between any two points. This will focus on all the ports involved in that route and can be an excellent way to focus in quickly on a performance or error situation which is being reported between 2 specific points in the fabric (Such as a **`StatusTimeoutRetry`** that MPI may be reporting between 2 processes in its run).

Focus can use glob style patterns. This permits a wildcarded focus by node name or IOC name. If a naming convention is used for fabric components, this can provide a powerful way to focus reports on nodes. For example, if the host names are prefixed with an indication of their purpose, searches can be performed based on the purpose of the node. For example if the following naming convention is used: **`l###`** = login node ###, **`n###`** = compute node ###, **`s###`** = storage node ###, etc. Node purposes can be focused by using patterns such as **`'l*'`**, **`'n*'`** or **`'s*'`**.

NOTE: A glob style pattern is a shell style wildcard pattern as used by **`bash`** and many other tools. When using such patterns they should be single quoted so that the shell will not try to expand them to match local file names.

Typically a focused report will include a summary at its start of the items focused on. When the focus has a large scope, this list can be quite long. In this case the **`-Q`** option can be used to omit this section from the report.

5.4.3.3.2
Focus Examples:

Below are some examples of using the focus options:

```
iba_report -o nodes -F portguid:0x00066a00a000447b
iba_report -o nodes -F nodeguid:0x00066a009800447b:port:1
iba_report -o nodes -F nodeguid:0x00066a009800447b
iba_report -o nodes -F node:duster
iba_report -o nodes -F node:duster:port:1
iba_report -o nodes -F 'nodepat:d*'
iba_report -o nodes -F 'nodepat:d*:port:1'
iba_report -o nodes -F nodetype:CA
iba_report -o nodes -F nodetype:CA:port:1
iba_report -o nodes -F lid:1
iba_report -o nodes -F gid:0xfe80000000000000:0x00066a00a000447b
iba_report -o nodes -F systemguid:0x00066a009800447b
iba_report -o nodes -F systemguid:0x00066a009800447b:port:1
iba_report -o nodes -F iocguid:0x00066a01300001e0
iba_report -o nodes -F iocguid:0x00066a01300001e0:port:2
iba_report -o nodes -F 'ioc:Chassis 0x00066A005000010C, Slot 2, IOC 1'
iba_report -o nodes -F 'ioc:Chassis 0x00066A005000010C, Slot 2, IOC 1:port:2'
iba_report -o nodes -F 'iocpat:*Slot 2*'
iba_report -o nodes -F 'iocpat:*Slot 2*:port:2'
iba_report -o nodes -F ioctype:VNIC
iba_report -o nodes -F ioctype:VNIC:port:2
iba_report -o nodes -F sm
iba_report -o nodes -F route:node:duster:node:cuda
iba_report -o nodes -F route:node:duster:port:1:node:cuda:port:2
```

5.4.3.4
Scriptable output

iba_report permits custom scripting. As previously mentioned, options like **-H**, **-P** and **-N** can aid the generation of reports that can be diff'ed.

In addition the **-x** option permits output reports to be generated in XML format. The XML hierarchy is similar to the textual reports. Use of XML permits other XML tools (such as PERL XML extensions) to easily parse **iba_report** output such that scripts can be created to further search and refine report output formats.

This allows **iba_report** to be integrated into custom scripts. It can also be used to generate customer-specific new report formats, cross reference **iba_report** with other site-specific information, etc.

5.4.3.4.1

Using iba_report to monitor for fabric changes

iba_report can easily be used in other scripts. For example the following simple script could be run as a cron job to identify if the fabric has changed as compared to the initial design:

```
#!/bin/bash
# specify some filenames to use
expected_config=/usr/local/report.master # master copy of
config previously created
config=/tmp/report$$ # where we will generate new report
diffs=/tmp/report.diff$$ # where we will generate diffs

iba_report -o all -d 5 -P > $config 2>/dev/null
if ! diff $config $expected_config > $diffs 2>/dev/null
then
   # notify admin, for example mail the new report to the admin
   cat $diffs $expected_config $config |
mail -s "fabric change detected" admin@somewhere
fi
rm -f $config $diffs
```

5.4.3.5
Sample Output

5.4.3.5.1
Analysis of all ports in fabric for errors, inconsistent connections, bad cables

```
[root@duster root]# iba_report -o errors -o slowconnlinks
Links running slower than faster port Summary

Links running slower than expected:
20 of 20 Links Checked, 0 Errors found

Links configured to run slower than supported:
Rate NodeGUID          Port Type Name
        Enabled      Supported
2.5g 0x00066a0098000384   1 CA cuda
      1x   2.5Gb   1-4x   2.5Gb
<->  0x00066a00d8000123   2 SW InfiniCon Systems InfinIO9024
    1-4x   2.5Gb   1-4x   2.5Gb
20 of 20 Links Checked, 1 Errors found

Links connected with mismatched speed potential:
20 of 20 Links Checked, 0 Errors found
-------------------------------------------------------------
Links with errors > threshold Summary

Configured Error Thresholds:
    SymbolErrorCounter             100
    LinkErrorRecoveryCounter       3
    LinkDownedCounter              3
    PortRcvErrors                  100
    PortRcvRemotePhysicalErrors    100
    PortXmitDiscards               100
    PortXmitConstraintErrors       10
    PortRcvConstraintErrors        10
    LocalLinkIntegrityErrors       3
```

```
    ExcessiveBufferOverrunErrors   3
    VL15Dropped                    100
Rate NodeGUID          Port Type Name
 10g 0x00066a0098000001   1 CA julio
<->  0x00066a00d8000123   8 SW InfiniCon Systems InfinIO9024
        LinkDownedCounter: 5 Exceeds Threshold: 3

 10g 0x00066a00980001b8   1 CA orc
<->  0x00066a00d8000123  10 SW InfiniCon Systems InfinIO9024
        LinkDownedCounter: 5 Exceeds Threshold: 3

 10g 0x00066a0098000380   1 CA goblin
        SymbolErrorCounter: 65535 Exceeds Threshold: 100
        LinkErrorRecoveryCounter: 255 Exceeds Threshold: 3
        PortRcvErrors: 65535 Exceeds Threshold: 100
<->  0x00066a00d8000123  15 SW InfiniCon Systems InfinIO9024
        SymbolErrorCounter: 41079 Exceeds Threshold: 100
        LinkErrorRecoveryCounter: 188 Exceeds Threshold: 3

 10g 0x00066a0098003f81   1 CA ibm345
<->  0x00066a00d8000123  12 SW InfiniCon Systems InfinIO9024
        SymbolErrorCounter: 9533 Exceeds Threshold: 100
        LinkErrorRecoveryCounter: 46 Exceeds Threshold: 3
        PortRcvErrors: 617 Exceeds Threshold: 100
20 of 20 Links Checked, 4 Errors found
```

5.4.3.5.2
Identification of the route between 2 nodes in the fabric

```
[root@duster root]# ./iba_report -o route -S node:orc -D
node:julio
Routes Summary Between:
    Node: 0x00066a00980001b8 CA orc
and Node: 0x00066a0098000001 CA julio

Routes between ports:
     0x00066a00980001b8   1 CA orc
and  0x00066a0098000001   1 CA julio
1 Paths
    SGID: 0xfe80000000000000:00066a00a00001b8
    DGID: 0xfe80000000000000:00066a00a0000001
    SLID: 0x000b DLID: 0x000c Reversible: Y PKey: 0xffff
    Raw: N FlowLabel: 0x00000 HopLimit: 0x00 TClass: 0x00
    SL:  0 Mtu: 2048 Rate:  10g PktLifeTime:   67 ms Pref: 0
        Rate NodeGUID          Port Type Name
         10g 0x00066a00980001b8   1 CA orc
     ->  0x00066a00d8000123  10 SW InfiniCon Systems InfinIO9024
      10g 0x00066a00d8000123  8 SW InfiniCon Systems InfinIO9024
        ->   0x00066a0098000001   1 CA julio
    2 Links Traversed
```

5.4.3.5.3

Analysis of the route between 2 nodes for errors, inconsistent connections, etc

```
[root@duster root]# ./iba_report -o errors -o slowconnlinks -Froute:node:orc:no e:julio
Links running slower than faster port Summary Focused on:
    4 Ports:
          1 0x00066a00a00001b8
            in Node: 0x00066a00980001b8 CA orc
         10
            in Node: 0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
          8
            in Node: 0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
          1 0x00066a00a0000001
            in Node: 0x00066a0098000001 CA julio

Links running slower than expected:
2 of 20 Links Checked, 0 Errors found

Links configured to run slower than supported:
2 of 20 Links Checked, 0 Errors found

Links connected with mismatched speed potential:
2 of 20 Links Checked, 0 Errors found
---------------------------------------------------------------
Links with errors > threshold Summary
Focused on:
    4 Ports:
          1 0x00066a00a00001b8
            in Node: 0x00066a00980001b8 CA orc
         10
            in Node: 0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
          8
            in Node: 0x00066a00d8000123 SW InfiniCon Systems InfinIO9024
          1 0x00066a00a0000001
            in Node: 0x00066a0098000001 CA julio

Configured Error Thresholds:
    SymbolErrorCounter             100
    LinkErrorRecoveryCounter       3
    LinkDownedCounter              3
    PortRcvErrors                  100
    PortRcvRemotePhysicalErrors    100
    PortXmitDiscards               100
    PortXmitConstraintErrors       10
    PortRcvConstraintErrors        10
    LocalLinkIntegrityErrors       3
    ExcessiveBufferOverrunErrors   3
    VL15Dropped                    100
Rate NodeGUID          Port Type Name
 10g 0x00066a0098000001   1 CA julio
<->  0x00066a00d8000123   8 SW InfiniCon Systems InfinIO9024
        LinkDownedCounter: 5 Exceeds Threshold: 3

 10g 0x00066a00980001b8   1 CA orc
<->  0x00066a00d8000123  10 SW InfiniCon Systems InfinIO9024
        LinkDownedCounter: 5 Exceeds Threshold: 3
2 of 20 Links Checked, 2 Errors found
```

5.4.3.5.4

Obtain very detailed information about nodes

NOTE: To shorten the length of the output, the following example focuses on only 1 node.

```
[root@duster root]# iba_report -o nodes -F node:erik -d 5 -s
Node Type Summary Focused on:
    System: 0x00066a0098004a73
        Node: 0x00066a00980003a6 CA erik
        Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:
    Name: erik
        NodeGUID: 0x00066a00980003a6 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    2 Connected Ports:
        PortNum:   1 LID: 0x0015 GUID: 0x00066a00a00003a6
            Neighbor:  0x00066a00d8000123   9 SW InfiniCon Systems InfinIO9024
            PortState: Active           PhysState: LinkUp    DownDefault: Pollg
            LID:    0x0015              LMC: 0               Subnet: 0xfe800000000000
            SMLID:  0x0001   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:6 ms
            M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
            MTU:       Active:    2048  Supported:    2048  VL Stall: 0
            LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
            LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
            VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
            Capability 0x02010048: CR CM SL Trap
            Violations: M_Key:     0 P_Key:     0 Q_Key:     0
            ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
            P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
                Performance: Transmit
                    Xmit Data                16383 MB (4294967295 Quads)
                    Xmit Pkts              431705985
                Performance: Receive
                    Rcv Data                 16383 MB (4294967295 Quads)
                    Rcv Pkts               424345537
                Errors:
                    Symbol Errors                  0
                    Link Error Recovery            0
                    Link Downed                    0
                    Port Rcv Errors                0
                    Port Rcv Rmt Phys Err          0
                    Port Rcv Sw Relay Err          0
                    Port Xmit Discards             0
                    Port Xmit Constraint           0
                    Port Rcv Constraint            0
                    Local Link Integrity           0
                    Exc. Buffer Overrun            0
                    VL15 Dropped                   0
        PortNum:   2 LID: 0x0016 GUID: 0x00066a01a00003a6
            Neighbor:  0x00066a00d8000123   7 SW InfiniCon Systems InfinIO9024
            PortState: Active           PhysState: LinkUp    DownDefault: Pollg
            LID:    0x0016              LMC: 0               Subnet: 0xfe800000000000
            SMLID:  0x0001   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:6 ms
            M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
            MTU:       Active:    2048  Supported:    2048  VL Stall: 0
            LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
            LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
            VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
            Capability 0x02010048: CR CM SL Trap
            Violations: M_Key:     0 P_Key:     0 Q_Key:     0
            ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
            P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
                Performance: Transmit
                    Xmit Data                      0 MB (0 Quads)
                    Xmit Pkts                      0
                Performance: Receive
                    Rcv Data                       0 MB (0 Quads)
                    Rcv Pkts                       0
                Errors:
                    Symbol Errors                  0
```

```
                    Link Error Recovery            0
                    Link Downed                    0
                    Port Rcv Errors                0
                    Port Rcv Rmt Phys Err          0
                    Port Rcv Sw Relay Err          0
                    Port Xmit Discards             0
                    Port Xmit Constraint           0
                    Port Rcv Constraint            0
                    Local Link Integrity           0
                    Exc. Buffer Overrun            0
                    VL15 Dropped                   0
    Name: erik
        NodeGUID: 0x00066a0098004a73 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    1 Connected Ports:
        PortNum:   1 LID: 0x0009 GUID: 0x00066a00a0004a73
            Neighbor:  0x00066a00d8000123  18 SW InfiniCon Systems InfinIO9024
            PortState: Active           PhysState: LinkUp    DownDefault: Pollg
            LID:    0x0009              LMC: 0               Subnet: 0xfe800000000000
            SMLID:  0x0001   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:6 ms
            M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
            MTU:       Active:    2048  Supported:    2048  VL Stall: 0
            LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
            LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
            VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
            Capability 0x02010048: CR CM SL Trap
            Violations: M_Key:     0 P_Key:     0 Q_Key:     0
            ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
            P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
                Performance: Transmit
                    Xmit Data                  16383 MB (4294967295 Quads)
                    Xmit Pkts              507314529
                Performance: Receive
                    Rcv Data                   16383 MB (4294967295 Quads)
                    Rcv Pkts               504418808
                Errors:
                    Symbol Errors                  0
                    Link Error Recovery            0
                    Link Downed                    0
                    Port Rcv Errors                0
                    Port Rcv Rmt Phys Err          0
                    Port Rcv Sw Relay Err          0
                    Port Xmit Discards             0
                    Port Xmit Constraint           0
                    Port Rcv Constraint            0
                    Local Link Integrity           0
                    Exc. Buffer Overrun            0
                    VL15 Dropped                   0
2 Matching CAs Found

3 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found
```

5.4.3.5.5

Obtain very detailed information about IOUs

NOTE: To shorten the length of the output, the following example focuses on only 1 IOC.

```
[root@duster root]# iba_report -o ious -F ioc:'Chassis 0x00066A005000010C, Slot 2, IOC 2' -d 5
IOU Summary Focused on:
    Ioc:   2 0x00066a02300001e0 Chassis 0x00066A005000010C, Slot 2, IOC 2
        in Node: 0x00066a00580001e0 CA VEx in Chassis 0x00066a005000010c, Slot

1 IOUs in Fabric:
    Name: VEx in Chassis 0x00066a005000010c, Slot 2
        NodeGUID: 0x00066a00580001e0 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a00580001e0
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    1 Connected Ports:
        PortNum:   2 LID: 0x0013 GUID: 0x00066a02580001e0
            Neighbor:  0x00066a00280002cd   3 SW InfiniCon Systems InfiniFabric
(Sw A Dev A)
            PortState: Active           PhysState: LinkUp    DownDefault: Pollig
            LID:    0x0013              LMC: 0               Subnet: 0xfe8000000000000
            SMLID:  0x0001   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:  56 ms
            M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
            MTU:       Active:    2048  Supported:    2048  VL Stall: 0
            LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
            LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
            VLs:       Active:     1+1  Supported:     4+1  HOQLife: 4096 ns
            Capability 0x02090048: CR DM CM SL Trap
            Violations: M_Key:     0 P_Key:     0 Q_Key:     0
            ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
            P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
    Max IOCs:   3 Change ID:     9 DiagDeviceId: 0 Rom: 0
        IocSlot:   2 GUID: 0x00066a02300001e0
            ID String: Chassis 0x00066A005000010C, Slot 2, IOC 2
            IO Class: 2000 SubClass: 66a Protocol: 0 Protocol Ver: 1
            VendorID: 0x66a DeviceID: 0x30 Rev: 0x1
            Subsystem: VendorID: 0x66a DeviceID: 0x30
            Capability: 0x33: ST SF WT WF
            Send Depth: 2 Size: 256; RDMA Read Depth: 0 RDMA Size: 4294967295
        2 Services:
            Name: InfiniNIC.InfiniConSys.Control:02
            Id: 0x1000066a00000002
            Name: InfiniNIC.InfiniConSys.Data:02
            Id: 0x1000066a00000102
1 Matching IOUs Found
```

5.4.3.5.6

Identify connections and links composing the fabric

```
[root@duster root]# iba_report -o links
Link Summary


20 Links in Fabric:
Rate NodeGUID          Port Type Name
 10g 0x00066a00280002cd   3 SW InfiniCon Systems InfiniFabric (Sw A Dev A)
<->  0x00066a00580001e0   2 CA VEx in Chassis 0x00066a005000010c, Slot 2
 10g 0x00066a00280002cd   5 SW InfiniCon Systems InfiniFabric (Sw A Dev A)
<->  0x00066a10280002cd   4 SW InfiniCon Systems InfiniFabric (Sw A Dev B)
 10g 0x00066a0098000001   1 CA julio
<->  0x00066a00d8000123   8 SW InfiniCon Systems InfinIO9024
```

```
 10g 0x00066a00980001b8   1 CA orc
<->  0x00066a00d8000123  10 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098000380   1 CA goblin
<->  0x00066a00d8000123  15 SW InfiniCon Systems InfinIO9024
2.5g 0x00066a0098000384   1 CA cuda
<->  0x00066a00d8000123   2 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098000384   2 CA cuda
<->  0x00066a00d8000123   1 SW InfiniCon Systems InfinIO9024
 10g 0x00066a00980003a6   1 CA erik
<->  0x00066a00d8000123   9 SW InfiniCon Systems InfinIO9024
 10g 0x00066a00980003a6   2 CA erik
<->  0x00066a00d8000123   7 SW InfiniCon Systems InfinIO9024
 10g 0x00066a00980006a2   1 CA goblin
<->  0x00066a00d8000123  20 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098000849   2 CA rockaway
<->  0x00066a00d8000123   3 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098002813   1 CA brady
<->  0x00066a00d8000123  19 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098002813   2 CA brady
<->  0x00066a00d8000123   5 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098002854   1 CA brady
<->  0x00066a00d8000123  11 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098002854   2 CA brady
<->  0x00066a00d8000123   6 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098003f81   1 CA ibm345
<->  0x00066a00d8000123  12 SW InfiniCon Systems InfinIO9024
 10g 0x00066a009800447b   1 CA duster
<->  0x00066a00d8000123   4 SW InfiniCon Systems InfinIO9024
 10g 0x00066a009800447b   2 CA duster
<->  0x00066a00d8000123  16 SW InfiniCon Systems InfinIO9024
 10g 0x00066a0098004a73   1 CA erik
<->  0x00066a00d8000123  18 SW InfiniCon Systems InfinIO9024
 10g 0x00066a00d8000123  14 SW InfiniCon Systems InfinIO9024
<->  0x00066a10280002cd   2 SW InfiniCon Systems InfiniFabric (Sw A Dev B)
```

5.4.3.5.7

Reverse lookups, translate a LID or GUID into the information about the node or port represented

```
[root@duster root]# iba_report -o nodes -F lid:5
Node Type Summary Focused on:
    Port:   1 0x00066a00a0000384
        in Node: 0x00066a0098000384 CA cuda

13 Connected CAs in Fabric:
    Name: cuda
        NodeGUID: 0x00066a0098000384 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098000384
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    2 Connected Ports:
        PortNum:   1 LID: 0x0005 GUID: 0x00066a00a0000384
           Neighbor:  0x00066a00d8000123   2 SW InfiniCon Systems InfinIO9024
            Width:   1x Speed:   2.5Gb
1 Matching CAs Found

3 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found
```

5.4.3.5.8

Forward lookups - lookup nodes or IOCs by name

```
[root@duster root]# iba_report -o nodes -F node:erik
Node Type Summary Focused on:
    System: 0x00066a0098004a73
        Node: 0x00066a00980003a6 CA erik
        Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:
    Name: erik
        NodeGUID: 0x00066a00980003a6 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    2 Connected Ports:
        PortNum:   1 LID: 0x0015 GUID: 0x00066a00a00003a6
            Neighbor:  0x00066a00d8000123   9 SW InfiniCon Systems InfinIO9024
            Width:   4x Speed:   2.5Gb
        PortNum:   2 LID: 0x0016 GUID: 0x00066a01a00003a6
            Neighbor:  0x00066a00d8000123   7 SW InfiniCon Systems InfinIO9024
            Width:   4x Speed:   2.5Gb
    Name: erik
        NodeGUID: 0x00066a0098004a73 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    1 Connected Ports:
        PortNum:   1 LID: 0x0009 GUID: 0x00066a00a0004a73
            Neighbor:  0x00066a00d8000123  18 SW InfiniCon Systems InfinIO9024
            Width:   4x Speed:   2.5Gb
2 Matching CAs Found

3 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found
```

5.4.3.5.9

Generate reports in a "comparible manner" so topology verification can be performed against a known good configuration

NOTE: To shorten the length of the output, the following example focuses on only 1 node.

```
[root@duster root]# iba_report -o nodes -F node:erik -d 5 -P
Node Type Summary Focused on:
    System: 0x00066a0098004a73
        Node: 0x00066a00980003a6 CA erik
        Node: 0x00066a0098004a73 CA erik

13 Connected CAs in Fabric:
    Name: erik
        NodeGUID: 0x00066a00980003a6 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    2 Connected Ports:
        PortNum:   1 LID: xxxxxx GUID: 0x00066a00a00003a6
            Neighbor:  0x00066a00d8000123   9 SW InfiniCon Systems InfinIO9024
            PortState: Active           PhysState: LinkUp    DownDefault: Pollig
            LID:    xxxxxx              LMC: 0               Subnet: 0xfe8000000000000
            SMLID:  xxxxxx   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:  56 ms
            M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
            MTU:       Active:    2048  Supported:    2048  VL Stall: 0
            LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
            LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
            VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
            Capability 0x02010048: CR CM SL Trap
            Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
            ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
            P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
        PortNum:   2 LID: xxxxxx GUID: 0x00066a01a00003a6
            Neighbor:  0x00066a00d8000123   7 SW InfiniCon Systems InfinIO9024
            PortState: Active           PhysState: LinkUp    DownDefault: Pollig
            LID:    xxxxxx              LMC: 0               Subnet: 0xfe8000000000000
            SMLID:  xxxxxx   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:  56 ms
            M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
            MTU:       Active:    2048  Supported:    2048  VL Stall: 0
            LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
            LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
            VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
            Capability 0x02010048: CR CM SL Trap
            Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
            ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
            P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
    Name: erik
        NodeGUID: 0x00066a0098004a73 Type: CA
        Ports: 2 PartitionCap: 64 SystemImageGuid: 0x00066a0098004a73
        BaseVer: 1 SmaVer: 1 VendorID: 0x66a DeviceID: 0x5a44 Rev: 0xa1
    1 Connected Ports:
        PortNum:   1 LID: xxxxxx GUID: 0x00066a00a0004a73
            Neighbor:  0x00066a00d8000123  18 SW InfiniCon Systems InfinIO9024
            PortState: Active           PhysState: LinkUp    DownDefault: Polli
g
```

```
                LID:    xxxxxx              LMC: 0               Subnet: 0xfe8000000000000
                SMLID:  xxxxxx   SMSL:  0   RespTimeout:   33 ms  SubnetTimeout:  56 ms
                M_KEY:  0x0000000000000000  Lease:     0 s       Protect: Readonly
                MTU:       Active:    2048  Supported:    2048  VL Stall: 0
                LinkWidth: Active:      4x  Supported:    1-4x  Enabled:    1-4x
                LinkSpeed: Active:   2.5Gb  Supported:   2.5Gb  Enabled:   2.5Gb
                VLs:       Active:     4+1  Supported:     4+1  HOQLife: 4096 ns
                Capability 0x02010048: CR CM SL Trap
                Violations: M_Key: xxxxx P_Key: xxxxx Q_Key: xxxxx
                ErrorLimits: Overrun: 15 LocalPhys: 15  DiagCode: 0x0000
                P_Key Enforcement: In: Off Out: Off  FilterRaw: In: Off Out: Off
2 Matching CAs Found

3 Connected Switches in Fabric:
0 Matching Switches Found

1 Connected SMs in Fabric:
0 Matching SMs Found
```

5.4.4
saquery

(All): **saquery** can perform various queries of the subnet manager/subnet agent and provide detailed fabric information.

In many cases **iba_report** provides a more powerful tool, however in some cases **saquery** is preferred, especially when dealing with service records and multicast.

Saquery is installed on all hosts as part of the IB stack, but it is also included in Fast Fabric. As such it can be a useful tool to run on the Fast Fabric host and is therefore also documented here.

By default **saquery** uses the first active port on the local system. However, if the IB management node is connected to more than one fabric (i.e., a subnet), the HCA and port may be specified to select the fabric whose SA is to be queried.

Usage:

```
saquery [-v] [-h hca] [-p port] [-o type]
              [-l lid] [-t type] [-s guid] [-n guid] [-g guid]
               [-u gid] [-m gid] [-d name]
               [-P 'guid guid'] [-G 'gid gid']
              [-a 'sguid...;dguid...'] [-A 'sgid...;dgid...']
```

Options:

-v/--verbose: verbose output

-h/--hca hca: hca to send via, default is 1st hca

-p/--port port: port to send via, default is 1st active port

-l/--lid lid: query a specific lid

-t/--type type: query by node type

-s/--sysguid guid: query by system image guid

-n/--nodeguid guid: query by node guid

-g/--portguid guid: query by port guid

-u/--portgid gid: query by port gid

-m/--mcgid gid: query by multicast gid

-d/--desc name: query by node name/description

-P/--guidpair 'guid guid': query by a pair of port Guids

-G/--gidpair 'gid gid': query by a pair of Gids

-a/--guidlist 'sguid ...;dguid ...': query by a list of port Guids

-A/--gidlist 'sgid ...;dgid ...': query by a list of Gids

-o/--output type: output type for query (default is node)

Node Types:

ca - channel adapter

sw - switch

rtr - router

GIDs:

Specify a 64 bit subnet and 64 bit interface ID as: **subnet:interface**. For example:

```
0xfe80000000000000:0x00066a00a0000380
```

Output Types:

systemguid: list of system image guids

nodeguid: list of node guids

portguid: list of port guids

lid: list of lids

gid: list of gids

desc: list of node descriptions/names

path: list of path records

node: list of node records

portinfo: list of port info records

sminfo: list of SM info records

swinfo: list of switch info records

link: list of link records

slvl: list of SL to VL mapping table records

vlarb: list of VL arbitration table records

pkey: list of P-Key table records

guids: list of GUID info records

service: list of service records

mcmember: list of multicast member records

inform: list of inform info records

linfdb: list of switch linear FDB records

ranfdb: list of switch random FDB records

mcfdb: list of switch multicast FDB records

trace: list of trace records

The following combinations of input (assorted **query by** options) and output (**-o**) are permitted:

Table 5-1. Input Combinations

Input option	-o output permitted	-o output not permitted
None	systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb,	gid, trace
-t node_type	systemguid, nodeguid, portguid, lid, desc, path, node	gid, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace
-l lid	systemguid, nodeguid, portguid, lid, desc, path, node, portinfo, swinfo, slvl, vlarb, pkey, guids, service, mcmember, linfdb, ranfdb, mcfdb	gid, sminfo, link, inform, trace
-s system_image_guid	systemguid, nodeguid, portguid, lid, desc, path, node	gid, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace
-n node_guid	systemguid, nodeguid, portguid, lid, desc, path, node	gid, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace
-g port_guid	systemguid, nodeguid, portguid, lid, desc, path, node, service, mcmember, inform, trace	gid, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, linfdb, ranfdb, mcfdb

Table 5-1. Input Combinations

Input option	-o output permitted	-o output not permitted
-u port_gid	path, service, mcmember, inform, trace	systemguid, nodeguid, portguid, lid, gid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, linfdb, ranfdb, mcfdb
-m multicast_gid	mcmember	systemguid, nodeguid, portguid, lid, gid, desc, path, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, inform, linfdb, ranfdb, mcfdb, trace
-d node_description	systemguid, nodeguid, portguid, lid, desc, path, node	Gid, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace
-P port_guid_pair	path, trace	systemguid, nodeguid, portguid, lid, gid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb
-G gid_pair	path, trace	systemguid, nodeguid, portguid, lid, gid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb
-a port_guid_list	path	systemguid, nodeguid, portguid, lid, gid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace
-A gid_list	path	systemguid, nodeguid, portguid, lid, gid, desc, node, portinfo, sminfo, swinfo, link, slvl, vlarb, pkey, guids, service, mcmember, inform, linfdb, ranfdb, mcfdb, trace

5.5 Advanced Initialization and Verification - ibtest

(Switch and Host) : **`Ibtest`** performs a number of multi-step operations. In general operations performed by ibtest involve a login to one or more target systems (hosts or SilverStorm IB chassis depending on options used). ibtest can perform software or firmware upgrades, reboot hosts, chassis and switches as well as perform a variety of host and fabric verification operations.

Usage:

```
ibtest [-cCn] [-i ipoib_suffix] [-f hostfile] [-F chassisfile]
[-h 'hosts'] [-H 'chassis'] [-N 'nodes'] [-L 'nodeFile'] [-r
release] [-I install_options] [-U upgrade_options] [-d dir] [-T
product] [-P packages] [-m netmask] [-a action] [-S] test ...
```

Options:

`-c`: clobber result files from any previous run before starting this run

`-C`: perform operation against chassis, default is **`hosts`**

`-n`: perform operation against IB node, default is **`hosts`**

`-i ipoib_suffix`; suffix to apply to host names to create ipoib host names. The default is **`-ib`**.

`-f hostfile`: file with hosts in cluster, default is **`/etc/sysconfig/iba/hosts`**

`-F chassisfile`: file with chassis in cluster. The default is **`/etc/sysconfig/iba/chassis`**

`-h hosts`: list of hosts to execute the command

`-H chassis`: list of chassis to execute the command

`-N nodes`: list of IB nodes to execute the command

`-L nodefile`: file with IB nodes in cluster. The default is **`/etc/sysconfig/iba/ibnodes`**

`-r release`: InfiniServ release to load/upgrade to, default is **`0trim`**

`-d dir`: directory to get InfiniServ.release.tgz from for load/upgrade

`-I install_options`: InfiniServ install options

`-U upgrade_options`: InfiniServ upgrade options

`-T product`: InfiniServ product type to install, default is **`InfiniServPerf`**. Other options include: **`InfiniServBasic`**, **`InfiniServMgmt`**, **`InfiniServTools`**.

`-P packages`: InfiniServ packages to install; default is **`iba ipoib mpi`**. The host allows: **`iba ibdev ibboot fastfabric ifibre inic ipoib`**

mpi mpidev mpisrc udapl sdp rds, or for a chassis upgrade, filenames/directories of firmware images to install. For directories specified, all **.pkg** files in directory tree will be used. **shell** wildcards may also be used within quotes, or for a switch upgrade, filename/directory of firmware image to install. For the directory specified, **.emfw** files in the directory tree will be used. **shell** wildcards may also be used within quotes.

-m netmask: IPoIB netmask to use for configipoib

-a action: action for firmware file for chassis upgrade

push: ensure firmware is in primary or secondary

select: ensure firmware is in primary

run: ensure firmware is in primary and running

The default is **push**.

-S: securely prompt for password for user on remote system/chassis test - test to run.

Host Test can be one or more of:

load: initial install of all hosts

upgrade: upgrade install of all hosts

configipoib: create ifcfg-ib1 using host IP addr from /etc/hosts

reboot: reboot hosts, ensure they go down and come back

sacache: confirm sacache has all hosts in it

ipoibping: verify this host can ping each host via IPoIB

mpiperf: verify latency and bandwitch for each host

Chassis Test can be one or more of:

reboot: reboot chassis, ensure they go down and come back

upgrade: upgrade install of all chassis

IB Node Test can be one or more of:

reboot: reboot IB node, ensure they go down and come back

upgrade: upgrade install of all IB nodes.

For example:

```
ibtest -c reboot
ibtest -h 'elrond arwen' reboot
HOSTS='elrond arwen' ibtest reboot
ibtest -C -a run -P '*.pkg' upgrade
```

Ibtest provides detailed logging of its results. During each run the following files are produced:

test.res: appended with summary results of run

test.log: appended with detailed results of run

save_tmp/: contains a directory per failed test with detailed logs

test_tmp*/: intermediate result files while test is running

The **-c** option will remove all of the above.

Results from **ibtest** are grouped into Test Suites, Test Cases and Test Items. A given run of **ibtest** represents a single Test Suite. Within a Test Suite multiple Test Cases will occur, typically one Test Case per host, chassis or switch being operated on. Some of the more complex operations (such as **ipoibping**) may have multiple Test Items per Test Case. Each such item represents a major step in the overall Test Case.

Each **ibtest** run appends to **test.res** and **test.log** and creates temporary files in **test_tmp$PID** in the current directory. **Test.res** will provide an overall summary of tests performed and their results. The same information will also be displayed while **ibtest** is executing. **Test.log** will contain detailed information about what was performed. This will include the specific commands executed and the resulting output. The **test_tmp** directories will contain temporary files which reflect tests in progress (or killed). The logs for any failures will be logged in the **save_temp** directory with a directory per failed test case. If the same test case fails more than once, **save_temp** will retain the information from the first failure and subsequent runs of **ibtest** will only append to **test.log**. It is recommended to review failures and use the **-c** option to remove old logs before subsequent runs of **ibtest**.

ibtest implicitly performs its operations in parallel. However, as for the other tools, **FF_MAX_PARALLEL** can be exported to change the degree of parallelism. Twenty (20) parallel operations is the default.

Environment Variables

The following environment variables are also used by this command:

HOSTS, HOSTS_FILE: see discussion on selection of hosts above

CHASSIS, CHASSIS_FILE: see discussion on selection of chassis above

IBNODES, IBNODES_FILE: see discussion on selection of switches above

FF_IPOIB_SUFFIX: suffix to append to hostname to create IPoIB hostname. Used in absence of **-i**

FF_MAX_PARALLEL: maximum number of parallel operations to perform at once.

FF_USERNAME: user name to login to hosts as, default is **root**

FF_PASSWORD: password to use to login as **FF_USERNAME**. Used in absence of -**S** option.

FF_ROOTPASS: password to use when **su** to **root** (if **FF_USERNAME** is not **root**). Used in absence of **-S** option.

FF_LOGIN_METHOD: how to login to hosts (Telnet, RSH or SSH), default is SSH

FF_TIMEOUT_MULT: multiplier for response timeouts. Default is 2. This typically does not need to be set, but in the event of unexpected timeouts or extremely slow hosts or chassis or management network, a larger value can be used.

FF_CHASSIS_LOGIN_METHOD: how to login to chassis. Can be SSH or Telnet

FF_CHASSIS_ADMIN_PASSWORD: password for admin on all chassis. Used in absence of **-S** option.

FF_PRODUCT: during host install and upgrade, what product should be used for installation (**InfiniServPerf**, **InfiniServBasic**, etc)

FF_INSTALL_OPTIONS: installation options for host IB software **INSTALL** during host "load". Used in absence of **-I** option.

FF_UPGRADE_OPTIONS: upgrade options for host IB software **INSTALL** during host "upgrade". Used in absence of **-U** option.

FF_PACKAGES: host packages to load during host "load". Directories or **.pkg** files to load during chassis "upgrade". Directories or **.emfw** files to load during switch "upgrade". Used In absence of **-P** option.

FF_IPOIB_NETMASK: netmask to use for IPoIB IP address during **configipoib**

5.5.1 ibtest Host Operations

(Host): It is recommended to set up password SSH or SCP for use during this operation. Alternatively, the **-S** option can be used to securely prompt for a password, in which case the same password is used for all hosts. Alternately, the password may be put in the environment or the **fastfabric.conf** file via **FF_PASSWORD** and **FF_ROOTPASS**.

5.5.1.1 load

This performs an initial installation of Fabric Access software on a group of hosts. Any existing Fabric Access installation will first be uninstalled and any Fabric Access configuration files will be removed. Therefore, the hosts will end up installed with a default Fabric Access configuration. The **-I** option can be used

to select different install packages, the defaults are **iba ipoib mpi** (i.e., IB Stack, IPoIB and MPI). The default is the typical configuration for an MPI cluster compute node. The **-r** option can be used to specify a release to install other than the one that this host is presently running. The **$FF_PRODUCT.<VERSION>.tgz** file (e.g., **InfiniServPerf.2.1.tgz**) is expected to exist in the directory specified by **-d** (the default is the current working directory) and will be copied to all the end nodes and installed.

5.5.1.2
upgrade

This is very similar to the **load** option, however all the nodes are upgraded without affecting existing Fabric Access configuration. This is comparible to the **-U** option when running **INSTALL** manually. The **-r** option can be used to upgrade to a release different from this host, the default will be to upgrade to the same release as the this host. The **$FF_PRODUCT.<VERSION>.tgz** file (e.g., **InfiniServPerf.2.1.tgz**) is expected to exist in the directory specified by **-d** (the default is the current working directory) and will be copied to all the end nodes and installed.

NOTE: Only those Fabric Access components that are currenly installed will be upgraded. This operation will fail for nodes that do not have Fabric Access software installed.

5.5.1.3
configipoib

Creates a **ifcfg-ib1** configuration file for each node using the IP address found via the resolver on the node (the standard Linux resolver is used via the **host** command). If the host is not found, **/etc/hosts** on the node is checked. The **-i** option can specify an IPoIB suffix to apply to the host name to create the IPoIB host name for the node (that will be looked up in **/etc/hosts**). The default suffix is **-ib**. The **-m** option can be used to specify a netmask other than the default for the given class of IP address (such as when dividing a class A or B address into smaller IP subnets). IPoIB will be configured for a static IP address and will be autostarted at boot. The default **/etc/sysconfig/ipoib.cfg** file will be used, which provides a redundant IPoIB configuration using both ports of the first HCA in the system.

5.5.1.4
reboot

This reboots the given hosts and ensures they go down and come back up by pinging them during the reboot process. The ping rate is slow (5 seconds), so if the servers boot faster than this, false failures may be seen.

5.5.1.5 sacache

This verifies the given hosts have properly communicated with the SA and cached paths to each other. To run this command, InfiniBand must be installed and running on the given hosts. The subnet manager and switches must be up. If this test fails, **`cmdall 'cat /proc/driver/ics_dsc/gids'`** can be run against any problem nodes to see what they have cached.

5.5.1.6 ipoibping

This verifies IPoIB basic operation by ensuring that the host can ping all other nodes via IPoIB. To run this command InfiniBand must be installed, IPoIB must be configured and running on the host and the given hosts, the SM and switches must be up. The **`-i`** option can specify an alternate IPoIB hostname suffix.

5.5.1.7 mpiperf

Verifies that MPI is operational and checks MPI end-to-end latency and bandwidth between pairs of nodes (e.g., 1-2, 3-4, 5-6). This can be used to verify switch latency/hops, PCI bandwidth and overall MPI performance. The **`test.res`** file will have the results of each pair of nodes tested.

To obtain accurate results, this test should be run at a time when no other stressful applications (e.g., MPI jobs, high stress filesystem operations) are running on the given hosts.

The following is a sample of expected MPI bandwidths for various server slot speeds:

- PCI-X 66 Mhz (32 bit) - 250 MB/s or less
- PCI-X 66Mhz - 400-450 MB/s or less
- PCI-X 100 Mhz - 600-700 MB/s
- PCI-X 133 Mhz - 800-900 MB/s
- PCIe x8 SDR HCA - 900+ MB/s
- PCIe x8 DDR HCA - 1300+ MB/s

Bandwidth issues typically indicate server configuration issues (e.g., incorrect slot used, incorrect BIOS settings, incorrect HCA model), or fabric issues (e.g., symbol errors, incorrect link width or speed). Assuming **`iba_report`** has previously been used to check for link errors and link speed issues, the server configuration should be verified.

Note that BIOS settings and differences between server models can account for 10-20% differences in bandwidth. The numbers above are conservative

numbers representative of what most servers can achieve. Some server models may have 10-20% higher results. A result 5-10% below the above numbers is typically not cause for serious alarm, but may reflect limitations in the server design or the chosen BIOS settings.

For more details about BIOS settings, consult the documentation from the server supplier and/or the server PCI chipset manufacturer.

5.5.2 ibtest Chassis Operations

(Switch): All operations against SilverStorm internally-managed IB chassis require the **-C** option to select chassis mode of operation for **ibtest**.

All chassis operations will login to the chassis as chassis user **admin**. It is recommended to use the **-S** option to securely prompt for a password, in which case the same password is used for all chassis. Alternately, the password may be put in the environment or the **fastfabric.conf** file via **FF_CHASSIS_ADMIN_PASSWORD**.

NOTE: Newer versions of chassis firmware permit SSH keys to be configured within the chassis for secure password-less login. In this case there is no need to configure a **FF_CHASSIS_ADMIN_PASSWORD** and **FF_CHASSIS_LOGIN_METHOD** can be SSH. Refer to the *SilverStorm 9000 Users Guide* for more information.

5.5.2.1 upgrade

This upgrades the firmware on each chassis or slot specified. The **-P** option selects a directory containing **.pkg** files or provides an explicit list of **.pkg** files for the chassis and/or slots. The **-a** option selects the desired minimal state for the new firmware. For each chassis and/or slot selected for upgrade, the **.pkg** file applicable to that slot will be selected and used. If more than one **.pkg** file is specified of a given card type, the operation is undefined.

The upgrade is intelligent and does not upgrade chassis that already have the desired firmware in the desired state (as specified by **-a**).

When the **-a** option specifies **run**, chassis that are not already running the desired firmware will be rebooted. By selecting the proper **FF_MAX_PARALLEL** value, a rolling upgrade or a parallel upgrade may be accomplished. In most cases a parallel upgrade is recommended for expediency.

For more information about chassis firmware refer to the *SilverStorm 9000 Users Guide* and *SilverStorm 9000 Release Notes*.

5.5.2.2
reboot

This reboots the given chassis and ensures they go down and come back up by pinging them during the reboot process.

By selecting the proper **`FF_MAX_PARALLEL`** value a rolling reboot or a parallel reboot may be accomplished. In most cases a parallel upgrade is recommended for expediency.

5.5.3
ibtest Switch Operations

(Switch): All operations against SilverStorm externally-managed switches require the **`-n`** option to select the switch mode of operation for **`ibtest`**.

5.5.3.1
reboot

Reboots the given switches.

By selecting the proper **`FF_MAX_PARALLEL`** value a rolling reboot or a parallel reboot may be accomplished. In most cases a parallel upgrade is recommended for expediency.

5.5.3.2
upgrade

Upgrades the firmware on each specified switch. The **`-P`** option selects a directory containing a **`.emfw`** file or provides an explicit **`.emfw`** file for the switches. If more than one **`.emfw`** file is specified, the operation is undefined. The **`-a`** option selects the desired minimal state for the new firmware. Only the **`select`** and **`run`** options are valid for this operation.

When the **`-a`** option specifies **`run`**, switches will be rebooted. By selecting the proper **`FF_MAX_PARALLEL`** value a rolling upgrade or a parallel upgrade may be accomplished. In most cases a parallel upgrade is recommended for expediency.

The upgrade process will also set the switch name. See discussion on selection of switches above.

For more information about switch firmware refer to the *SilverStorm 9000 Users Guide* and *SilverStorm 9000 Release Notes*.

5.5.4
Interpreting the ibtest log files

Each run of **ibtest** will create **test.log** and **test.res** files in the current directory.

When **ibtest** indicates that some or all of the test cases failed, the **test.res** and **test.log** files should be reviewed. **test.res** will summarize which tests have failed. Using the **test.res** file for servers that failed can be quickly identified. If the problem is not immediately obvious, check the **test.log** file. The most recent results will be at the end of the file. The **save_tmp/*/test.log** files will be easier to read since they will represent the logs for a single test case, typically against a single chassis, switch or host.

The keyword **FAILURE** will be used to mark any failures. Typically due to the roll up of error messages, the first instance of **FAILURE** in a given sequence of failures will show what was being done. Preceeding the **FAILURE**, the log will also show the exact sequence of commands issued to the target host and/or chassis and the resulting output from that host and/or chassis.

For example, **test.log** may contain lines such as:

```
scp ./InfiniServPerf.4.1.1.0.15.tgz root@n001a:


TEST CASE FAILURE=scp ./InfiniServPerf.4.1.1.0.15.tgz
root@n001a: failed: ssh: n001a Name or service not known

lost connection
```

This indicates the scp command shown was executed but failed with the error message **"ssh: n001a Name or service not known**.

lost connection"

In this example, this was the exact output from SSH.

If there is a **FAILURE** message indicating timeout, it means the expected output did not occur within a reasonable time limit. The time limits used are quite generous, so such failures often indicate a host, chassis or switch is offline. It could also indicate unexpected prompts (such as a password prompt when password-less ssh is expected). Review the test.log first for such prompts. Also verify that the host can SSH to the the target host or chassis with the expected password behavior.

Another common source of timeouts is incorrect host shell command prompts. Verify that both this host and the target host have their prompts set correctly. The command line prompt must end in **#** or **$** (make certain there is a space after either).

Yet another common source of timeouts is typograhical errors in selected host or chassis names. Verify that the host, chassis or switch names in **test.log**

match the intended host names. Also make sure than when IPoIB host names are used, that the correct name was formed based on the **ibtest -i '<IPOIB SUFFIX>'** argument. This applies a suffix to host names to create IPoIB host names. The default is **-ib**. Use **-i ''** to indicate no suffix.

5.6 Health Check and Baselining Tools

(All): These tools help to rapidly identify if the fabric has a problem or if its configuration has changed since the last baseline. Analysis includes hardware, software, fabric topology and SM configuration. The tools are designed to permit easy manual execution or automated execution via **cron** or other mechanisms.

These tools consist of 5 commands:

all_analysis - performs selected set of the below 4 analysis commands. This command is recommended as the primary tool for general analysis. When its desired to restrict the analysis to a specific subset of components, use one of the commands below.

fabric_analysis - performs fabric topology and PMA error counters analysis.

chassis_analysis - performs SilverStorm IB chassis configuration and health analysis for selected chassis.

esm_analysis - performs embedded SM configuration and health analysis for selected chassis.

hostsm_analysis - performs host SM configuration and health analysis for the local host.

5.6.1 Usage Model

These tools all support three modes of operation: health check only, baseline and check mode. The typical usage model for the tools is as follows:

- Perform initial fabric install and verification
  - Optionally run tools in "health check only" mode
  - Performs quick health check
  - Duplicates some of steps already done during verification
- Run tools in "baseline" mode
  - Takes a baseline of present HW/SW/config
- Periodically run tools in "check" mode
  - Performs quick health check

- Compares present HW/SW/config to baseline
- Can be scheduled in hourly cron jobs

- As needed rerun "baseline" when expected changes occur
  - Fabric upgrades
  - Hardware replacements/changes
  - Software configuration changes
  - Etc.

5.6.2
Common Operations and Options

The Health Check and Baselining tools support the following options to select the operations to be done by the tool:

-b - perform a baseline snapshot of the configuration

-e - perform an error check/health analysis only

If neither option is specified, the tool performs a snapshot of the present configuration, compares it to the baseline and also perform an error check/health analysis.

Use of both **-b** and **-e** on a given run is not permitted.

The typical use of the tools is to perform an initial error check by running the **-e** option. Review the errors reported in the files indicated by the tools. Once all the errors are corrected, perform a baseline of the configuration using the **-b** option. The baseline configuration will be saved to files in **$FF_ANALYSIS_DIR/baseline** (the default of **/var/opt/iba/analysis/baseline** is set via **/etc/sysconfig/fastfabric.conf**). This baseline configuration should be carefully reviewed to make sure it matches the intended configuration of the cluster. If it does not, the cluster should be corrected and a new baseline run.

For example:

```
fabric_analysis -e
```

errors reported could include links with high error rates, unexpected low speeds, etc. Correct any such errors then rerun **fabric_analysis -e** to make sure there is a good fabric.

```
fabric_analysis -b
```

The baseline configuration will be saved to **$FF_ANALYSIS_DIR/baseline**. This will include files starting with **links** and **comps**. These will be the results of **iba_report -o** links and **iba_report -o** comps reports respectively. Review these files and make sure all the expected links and components are present. For example, make sure all the switches and servers in the cluster are

present. Also verify the appropriate links between servers and switches are present. If the fabric is not correctly configured, correct the configuration and rerun the baseline.

Once a good baseline has been established, use the tools to compare the present fabric against the baseline and check its health. For example:

```
fabric_analysis
```

Will check the present fabric links and components against the previous baseline. If there have been changes, it will report a failure and indicate which files hold the resulting snapshot and differences. It will also check the PMA error counters and link speeds for the fabric (similar to **fabric_analysis -e**). If either of these checks fail, it will return a non-zero exit status, therefore permitting higher level scripts to detect a failed condition.

The differences files are generated using the Linux command specified by **FF_DIFF_CMD** in **fastfabric.conf.** By default this is the **diff -C 1** command. It is run against the baseline and new snapshot. Therefore, lines after each ***** #,# ****** heading in the **diff** are from the baseline and lines after each **--- #,# ----** heading are from the new snapshot. If **FF_DIFF_CMD** is simpy set to **diff**, lines indicated by **"<"** in the diff would be from the baseline and lines indicated by **">"** in the **diff** would be from the new snapshot. Another command which can be useful is the Linux **sdiff** command. For more information about the **diff** output format, consult the Linux man page for **diff**.

If the configuration is intentionally changed, a new error analysis and baseline should be obtained using the same sequence as for the initial installation (discussed above), establishing a new baseline for future comparisons.

In addition all of the tools support the following two options:

-s - save history of failures.

-d dir - top level directory for saving baseline, snapshots and history (default is **$FF_ANALYSIS_DIR** which is set in **fastfabric.conf**).

When the **-s** option is used, each failed run will also create a directory (whose name is the date/time the analysis tool was started) containing the failing snapshot information and **diffs**. This will permit a history of failures to be tracked. Note that every run of the tools also creates a **latest** directory with the latest snapshot. The **latest** files are overwritten by each subsequent run of the tool, which means the most recent run results are always available.

Beware, frequent use of the health check tools in conjunction with **-s** can consume a large amount of disk space. The space requirements will depend greatly on the size of the cluster, for example it could be > 10 megabytes per run on a 1000 node cluster.

The **-d** option allows command line control over the baseline, snapshot and history directory tree. Runs using **-d** must use the same directory as any

previous baseline which is to be compared to (except when **-e** option is used). The **FF_ANALYSIS_DIR** option in **fastfabric.conf** can be changed to provide a customer specific alternate directory which will be used whenever the **-d** option is not specified. Under **FF_ANALYSIS_DIR** subdirectories will be created as follows:

- **baseline** - baseline snapshot from each analysis tool.
- **latest** - latest snapshot from each analysis tool.
- **YYYY-MM-DD-HH:MM:SS** - failed analysis from analysis run with **-s**. Actual directory name will have actual date/time as the name.

5.6.3 fabric_analysis

(All): The fabric_analysis command has the following usage:

```
fabric_analysis [-b|-e] [-s] [-d dir] [-c file] [-t portsfile]
[-p ports]
```

-b - baseline mode, default is compare/check mode.

-e - evaluate health only, default is compare/check mode.

-s - save history of failures (errors/differences).

-d dir - top level directory for saving baseline and history of failed checks. The default is **/var/opt/iba/analysis**.

-c file - error thresholds config file. The default is **/etc/sysconfig/iba/iba_mon.conf**.

-t portsfile - file with list of local HCA ports used to access fabric(s) for analysis. The default is **/etc/sysconfig/iba/ports**.

-p ports - list of local HCA ports used to access fabric(s) for analysis. The default is the first active port.

This is specified as HCA:port:

0:0 = 1st active port in system

0:y = port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

for example:

```
fabric_analysis
fabric_analysis -p '1:1 1:2 2:1 2:2'
```

The fabric analysis tool checks the following:

- fabric links (both internal to switch chassis and external cables)

- fabric components (nodes, links, SMs, systems, and their SMA configuration)
- fabric PMA error counters and link speed mismatches

Note that the comparison includes components on the fabric. Therefore operations such as shutting down a server will cause the server to no longer appear on the fabric and will be flagged as a fabric change or failure by **fabric_analysis**.

Environment Variables:

The following environment variables are also used by this command:

PORTS - list of ports, used in absence of **-t** and **-p**.

PORTS_FILE - file containing list of ports, used in absence of **-t** and **-p**.

FF_ANALYSIS_DIR - top level directory for baselines and failed health checks.

FF_CURTIME - timestamp to use on directory created in **FF_ANALYSIS_DIR**, default is the present date and time.

FF_FABRIC_HEALTH - **iba_report** options to use during a health check.

FF_DIFF_CMD -Linux command used to compare the baseline to the latest snapshot.

For simple fabrics, the Fast Fabric host would be connected to a single fabric. By default the first active port on the Fast Fabric host will be used to analyze the fabric.

However, in more complex fabrics, the Fast Fabric host may be connected to more than one fabric (e.g., an IB subnet). In this case the specific ports and/or HCAs to use for fabric analysis may be specified. The HCA and port number specified will become part of the filenames in the **FF_ANALYSIS_DIR** such that unique status can be tracked for each fabric.

Specification of the ports to be used can be performed on the command line using the **-p** option, in a file specified via the **-t** option, via the environment variables **PORTS** or **PORTS_FILE**, or via the **PORTS_FILE** configuration option in **fastfabric.conf**. If the specified file does not exist or is empty, the first active port on the local system will be used. In more complex configurations (such as where the Fast Fabric host is connected to multiple IB fabrics or IB subnets), the user will need to specify the exact ports to use such that all fabrics are analyzed. For more information, refer to the section “Selection of local Ports (subnets)” on page 5-9.

By default the error analysis includes PMA counters and slow links (i.e., links running below enabled speeds). However this can be changed via the **FF_FABRIC_HEALTH** configuration parameter in **fastfabric.conf** (see appendix B for more information). This parameter specifies the **iba_report**

options and reports to be used for the health analysis. It also can specify the PMA counter clearing behavior (**-i seconds**, **-C**, or none at all).

The thresholds for PMA counter analysis default to **/etc/sysconfig/iba/iba_mon.conf**. However, an alternate configuration file for thresholds can be specified using the **-c** option.

All files generated by **fabric_analysis** will start with **fabric** in their file name. This is followed by the port selection option (default of **0:0**) identifying the port (i.e., IB subnet) used for the analysis.

The **fabric_analysis** tool generates files such as the following within **FF_ANALYSIS_DIR**:

Health Check:

latest/fabric.0:0.errors - **stdout** of **iba_report** for errors encountered during fabric error analysis.

latest/fabric.0.0.errors.stderr - **stderr** of **iba_report** during fabric error analysis.

Baseline:

baseline/fabric.0:0.comps - **iba_report** summary of fabric components and basic SMA configuration.

baseline/fabric.0.0.links - **iba_report** summary of internal and external links.

During a baseline run, the above files are also created in **FF_ANALYSIS_DIR/latest**.

Full analysis:

latest/fabric.0:0.errors - **stdout** of **iba_report** for errors encountered during fabric error analysis.

latest/fabric.0.0.errors.stderr - **stderr** of **iba_report** during fabric error analysis.

latest/fabric.0:0.comps - **stdout** of **iba_report** for fabric components and SMA configuration.

latest/fabric.0:0.comps.stderr - **stderr** of **iba_report** for fabric components.

latest/fabric.0:0.comps.diff - **diff** of baseline and latest fabric components.

latest/fabric.0:0.links - **stdout** of **iba_report** summary of internal and external links.

latest/fabric.0:0.links.stderr - **stderr** of **iba_report** summary of internal and external links.

latest/fabric.0:0.links.diff - **diff** of baseline and latest fabric internal and external links.

The **.diff** files are only created if differences are detected.

If the **-s** option is used and failures are detected, files related to the checks that failed are also copied to the timestamped directory name under **FF_ANALYSIS_DIR**, such as:

```
FF_ANALYSIS_DIR/2007-11-22-09:53:04
```

5.6.3.1

IB Fabric items checked against the baseline

- Based on iba_report -o links:
  - Unconnected/down/missing cables
  - Added/moved cables
  - Changes in link width and speed
  - Changes to Node GUIDs in fabric (replacement of HCA or IB Switch hardware)
  - Adding/Removing IB Nodes (CA, Virtual CAs, Virtual Switches, Physical Switches, Physical Switch internal switching cards (leaf/spine))
  - Changes to server or switch names
- Based on iba_report -o comps:
  - Overlap with items above from links report
  - Changes in port MTU, LMC, number of VLs
  - Changes in port speed/width enabled or supported
  - Changes in HCA or switch device IDs/revisions/VendorID (eg. ASIC HW changes)
  - Changes in port Capability mask (which IB features/agents run on port/server)
  - Changes to ErrorLimits and PKey enforcement per port
  - Changes to IOUs/IOCs/IOC Services provided
    - only applicable if IOUs in fabric (9000 series Virtual IO cards, IB native storage, etc)
- Location (port, node) and number of SMs in fabric
  - includes primary and backups
  - includes configured priority for SM

5.6.3.2
IB fabric Items that are also checked during healthcheck

Based on **`iba_report -o errors -o slowlinks`**:

- PMA error counters on all IB ports (HCA, switch external and switch internal) checked against configurable thresholds.
  - Counters are cleared each time a healthcheck is run, each healthcheck reflects a counter delta since last healthcheck.
  - Typically identifies potential fabric errors (symbol errors, etc).
  - May also identify transient congestion (depends upon counters monitored).
- Link active speed/width as compared to Enabled speed.
  - Identifies links whose active speed/width is < min(enabled speed/width on each side of link).
  - This typically reflects bad cables or bad ports or poor connections.
- Side effect is the verification of SA health.

5.6.4
chassis_analysis

(Switch): The **`chassis_analysis`** command has the following usage:

```
chassis_analysis [-b|-e] [-s] [-d dir] [-F chassisfile] [-H
'chassis']
```

`-b` - baseline mode. The default is the compare/check mode.

`-e` - evaluate health only, default is the compare/check mode.

`-s` - a save history of failures (errors/differences).

`-d dir` - the top level directory for saving baseline and history of failed checks. The default is **`/var/opt/iba/analysis`**.

`-F chassisfile` - the file with the chassis in the cluster. The default is **`/etc/sysconfig/iba/chassis`**.

`-H chassis` - a list of chassis on which to execute the command.

For example:

```
chassis_analysis
```

The chassis analysis tool checks the following for SilverStorm IB chassis:

- Chassis configuration (as reported by the chassis commands specified in **`FF_CHASSIS_CMDS`** in **`fastfabric.conf`**).
- Chassis health (as reported by the chassis command specified in **`FF_CHASSIS_HEALTH`** in **`fastfabric.conf`**).

Environment Variables

The following environment variables are also used by this command:

CHASSIS, **CHASSIS_FILE** - see the discussion on the selection of chassis above.

FF_TIMEOUT_MULT - multiplier for response timeouts. The default is 2. this typically does not need to be set, but in the event of unexpected timeouts or extremely slow chassis or management network, a larger value can be used.

FF_CHASSIS_LOGIN_METHOD - how to login to a chassis. Can be SSH or Telnet.

FF_CHASSIS_ADMIN_PASSWORD - The password for the administrator on all chassis. Used in absence of the **-S** option.

FF_CURTIME - The timestamp to use on a directory created in **FF_ANALYSIS_DIR**. The default is the present date and time.

FF_CHASSIS_CMDS - the set of chassis CLI commands to fetch chassis configuration information.

FF_CHASSIS_HEALTH - the chassis CLI command to check the chassis health.

FF_DIFF_CMD -Linux command used to compare the baseline to the latest snapshot.

The default set of **FF_CHASSIS_CMDS** is:

```
showInventory fwVersion showIBNodeDesc ismShowPStatThresh
ismChassisSet12x timeZoneConf timeDSTConf snmpCommunityConf
snmpTargetAddr showChassisIpAddr showDefaultRoute
```

The commands specified in **FF_CHASSIS_CMDS** must be simple commands with no arguments. The output of these commands will be textually compared (via **FF_DIFF_CMD**) to the baseline. Therefore, commands that include dynamically changing values (such as port packet counters) should not be included in this list.

FF_CHASSIS_HEALTH can specify one command (with arguments) to be used to check the chassis health. For chassis with newer firmware, the **hwCheck** command is recommended. For chassis with older firmware a benign command, such as **fruInfo**, should be used. The default is **hwCheck**. Note that only the exit status of the **FF_CHASSIS_HEALTH** command is checked. The output is not captured and compared in a snapshot. However, on failure its output is saved to aid diagnosis.

The **chassis_analysis** tool performs its analysis against one or more chassis in the fabric. As such, it permits the chassis to be specified via the **-H**, **-F**, **CHASSIS, CHASSIS_FILE** or **fastfabric.conf**. The handling of these options and settings is comparable to **cmdall -C** and similar Fast fabric commands against a chassis.

All files generated by **fabric_analysis** start with **chassis.** in the file name.

The **chassis_analysis** tool generates files such as the following within **FF_ANALYSIS_DIR**. The actual file names reflect the individual chassis commands that have been configured via the **FF_CHASSIS_HEALTH** and **FF_CHASSIS_CMDS** parameters:

Health Check:

latest/chassis.hwCheck - output of hwCheck command for all selected chassis

Baseline:

baseline/chassis.fwVersion - the output of **fwVersion** command for all selected chassis.

baseline/chassis.ismChassisSet12x - the output of the **ismChassisSet12x** command for all selected chassis.

baseline/chassis.ismShowPStatThresh - the output of the **ismShowPStatThresh** command for all selected chassis.

baseline/chassis.showChassisIpAddr -the output of the **showChassisIpAddr.** command for all selected chassis.

baseline/chassis.showDefaultRoute - the output of the **showDefaultRoute** command for all selected chassis.

baseline/chassis.showIBNodeDesc - the output of the **showIBNodeDesc** command for all selected chassis.

baseline/chassis.showInventory - the output of the **showInventory** command for all selected chassis.

baseline/chassis.snmpCommunityConf - the output of the **snmpCommunityConf** command for all selected chassis.

baseline/chassis.snmpTargetAddr - the output of the **snmpTargetAddr** command for all selected chassis.

baseline/chassis.timeDSTConf - the output of the **timeDSTConf** command for all selected chassis.

baseline/chassis.timeZoneConf - the output of the **timeZoneConf** command for all selected chassis.

During a baseline run, the above files are also created in **FF_ANALYSIS_DIR/latest**.

Full analysis:

latest/chassis.hwCheck - the output of the **hwCheck** command for all selected chassis.

latest/chassis.fwVersion - the output of the **fwVersion** command for all selected chassis.

latest/chassis.fwVersion.diff - **diff** of the baseline and latest **fwVersion**.

latest/chassis.ismChassisSet12x - the output of the **ismChassisSet12x** command for all selected chassis.

latest/chassis.ismChassisSet12x.diff - the **diff** of the baseline and latest **ismChassisSet12x**.

latest/chassis.ismShowPStatThresh - the output of the **ismShowPStatThresh** command for all selected chassis.

latest/chassis.ismShowPStatThresh.diff - the **diff** of baseline and latest **ismShowPStatThresh**.

latest/chassis.showChassisIpAddr - the output of the *showChassisIpAddr* command for all selected chassis.

latest/chassis.showChassisIpAddr.diff - the **diff** of baseline and latest **showChassisIpAddr**.

latest/chassis.showDefaultRoute - the output of the **showDefaultRoute** command for all selected chassis.

latest/chassis.showDefaultRoute.diff - the **diff** of the baseline and the latest **showDefaultRoute**.

latest/chassis.showIBNodeDesc - the output of the **showIBNodeDesc** command for all selected chassis.

latest/chassis.showIBNodeDesc.diff - the **diff** of the baseline and latest **showIBNodeDesc**.

latest/chassis.showInventory - the output of the **showInventory** command for all selected chassis.

latest/chassis.showInventory.diff - **diff** of the baseline and latest **showInventory**.

latest/chassis.snmpCommunityConf - the output of the **snmpCommunityConf** command for all selected chassis.

latest/chassis.snmpCommunityConf.diff - **diff** of the baseline and latest **snmpCommunityConf**.

latest/chassis.snmpTargetAddr - the output of the **snmpTargetAddr** command for all selected chassis.

latest/chassis.snmpTargetAddr.diff - **diff** of the baseline and latest **snmpTargetAddr**.

latest/chassis.timeDSTConf - the output of the **timeDSTConf** command for all selected chassis.

latest/chassis.timeDSTConf.diff - diff of the baseline and latest **timeDSTConf**.

latest/chassis.timeZoneConf - the output of the **timeZoneConf** command for all selected chassis.

latest/chassis.timeZoneConf.diff - diff of the baseline and latest **timeZonfConf**.

The **.diff** files are only created if differences are detected.

If the **-s** option is used and failures are detected, files related to the checks that failed are also copied to a time stamped directory name under **FF_ANALYSIS_DIR**, such as:

```
FF_ANALYSIS_DIR/2007-11-22-09:53:04
```

5.6.4.1
Chassis items checked against the baseline

Based upon **showInventory**:

- addition/removal of Chassis FRUs
  - Replacement is only checked for FRUs that **showInventory** displays the serial number. For the 9000 series, the fan and power supply replacement is not checked, just present.
- removal of redundant FRUs (spines, power supply, fan)

Based upon **fwVersion**:

- changes to primary or alternate FW versions installed in cards in chassis

Based upon **showIBNodeDesc**:

- changes to configured IB node description for chassis. Note changes detected here would also be detected in fabric level analysis

Based upon **ismShowPStatThresh**:

- changes to configured port thresholds for chassis port error thresholding

Based upon **ismChassisSet12x**:

- changes to chassis link width controls. Note that changes detected here would also be detected in fabric level analysis.

Based upon **timeZoneConf** and **timeDSTConf**:

- changes to the chassis time zone and daylight savings time configuration

Based upon **snmpCommunityConf** and **snmpTargetAddr**:

- changes to SNMP persistent configuration within the chassis

The following Chassis items will not be checked against baseline:

- changes to the chassis configuration on the management LAN (e.g., **showChassisIpAddr**, **showDefaultRoute**). Such changes will typically result in the chassis not responding on the LAN at the expected address that is detected by failures that will perform other chassis checks.

5.6.4.2
Chassis Items also checked during healthcheck

Based upon **hwCheck**:

- Overall health of FRUs in chassis:
  - Status of Fans in chassis
  - Status of Power Supplies in chassis
  - Temp/Voltage for each card
- Presence of adequate power/cooling of FRUs
- Presence of N+1 power/cooling of FRUs
- Presence of Redundant AC input

5.6.5
hostsm_analysis

(All): The **hostsm_analysis** command has the following usage:

```
hostsm_analysis [-b|-e] [-s] [-d dir]
```

-b - baseline mode. The default is the compare/check mode.

-e - evaluate health only. The default is the compare/check mode.

-s - save history of failures (i.e., errors/differences).

-d dir - the top level directory for saving baseline and history of failed checks. The default is **/var/opt/iba/analysis**.

for example:

```
hostsm_analysis
```

The host SM analysis tool checks the following:

- host SM software version
- host SM configuration file (simple text compare via **FF_DIFF_CMD**)
- host SM health (i.e., is it running?)

The **hostsm_analysis** tool performs analysis against the local server only. It is assumed that both the host SM and Fast Fabric are installed on the same system.

Environment Variables

The following environment variables are also used by this command:

FF_CURTIME - the timestamp to use on the directory created in
FF_DIFF_CMD - Linux command to use to compare baseline to latest snapshot

All files generated by **hostsm_analysis** start with **hostsm.** in the file name.

The **hostsm_analysis** tool generates files such as the following within **FF_ANALYSIS_DIR**. The actual file names reflect the individual chassis commands that have been configured via the **FF_CHASSIS_HEALTH** and **FF_CHASSIS_CMDS** parameters:

Health Check:

latest/hostsm.smstatus - the output of the **sm_query smShowStatus** command.

Baseline:

baseline/hostsm.smver - host SM version.

baseline/hostsm.smconfig - a copy of **iview_fm.config**.

During a baseline run, the above files are also created in **FF_ANALYSIS_DIR/latest**.

Full analysis:

latest/hostsm.smstatus - the output of the **sm_query smShowStatus** command.

latest/hostsm.smver - the host SM version.

latest/hostsm.smver.diff - **diff** of the baseline and latest host SM version.

latest/hostsm.smconfig - a copy of **iview_fm.config**.

latest/hostsm.smconfig.diff - **diff** of the baseline and the latest **iview_fm.config**.

The **.diff** files are only created if differences are detected.

If the **-s** option is used and failures are detected, files related to the checks that failed are also copied to a time stamped directory name under *FF_ANALYSIS_DIR*, such as:

```
FF_ANALYSIS_DIR/2007-11-22-09:53:04
```

5.6.5.1
Host SM items checked against the baseline

- SM configuration file
- The version of the SM rpm installed on the system

5.6.5.2
Host SM items also checked during healthcheck

- The SM is in the running state

5.6.6
esm_analysis

(Switch): The **esm_analysis** command has the following usage:

```
esm_analysis [-b|-e] [-s] [-d dir] [-G esmchassisfile] [-E
'esmchassis']
```

-b - baseline mode. The default is the compare/check mode.

-e - evaluate health only. The default is the compare/check mode.

-s - save history of failures (i.e., errors/differences).

-d dir - the top level directory for saving baseline and history of failed checks. The default is **/var/opt/iba/analysis**.

-G esmchassisfile - the file with SM chassis within the cluster. The default is **/etc/sysconfig/iba/esm_chassis**.

-E esmchassis - the list of the SM chassis to be analyzed.

for example:

```
esm_analysis
```

The embedded SM analysis tool checks the following:

- embedded SM configuration (as reported by the chassis commands specified in **FF_ESM_CMDS** in **fastfabric.conf**).
- embedded SM health (as reported by **smControl status**).

Environment Variables

The following environment variables are also used by this command:

ESM_CHASSIS, ESM_CHASSIS_FILE - see the discussion on the selection of chassis above. These have the same format as **CHASSIS** and **CHASSIS_FILE**.

FF_TIMEOUT_MULT - a multiplier for response timeouts. The default is 2. This typically does not need to be set, but in the event of unexpected timeouts

or extremely slow chassis or management network, a larger value can be used.

FF_CHASSIS_LOGIN_METHOD - how to login to a chassis. Can be SSH or Telnet.

FF_CHASSIS_ADMIN_PASSWORD - the password for administrator on all chassis. Used in absence of the **-S** option.

FF_CURTIME - The time stamp to use on the directory created in **FF_ANALYSIS_DIR**. The default is the current date and time.

FF_ESM_CMDS - a set of chassis CLI commands to get the SM configuration.

FF_DIFF_CMD - a Linux command to use to compare baseline to latest snapshot.

The default set of **FF_ESM_CMDS** is:

```
smShowSMParms smShowDefBcGroup
```

The commands specified in **FF_ESM_CMDS** must be simple commands with no arguments. The output of these commands will be textually compared (via **diff**) to the baseline. Therefore, commands that include dynamically changing values (such as port packet counters) should not be included in this list.

The **esm_analysis** variable performs analysis against one or more chassis in the fabric. As such it permits a chassis to be specified via the **-E**, **-G**, **ESM_CHASSIS, ESM_CHASSIS_FILE** or **fastfabric.conf**. The handling of these options and settings is comparible to **cmdall -C** and similar Fast fabric commands against a chassis. The exception in this case is that the option and variable names are slightly different to distinguish the fact theat they are specifying only the chassis that has an embedded SM running).

All files generated by **esm_analysis** start with **esm** within the file name.

The **esm_analysis** variable generates files such as the following within **FF_ANALYSIS_DIR**. The actual file names reflect the individual chassis commands that have been configured via the **FF_ESM_CMDS** parameter:

Health Check:

latest/esm.smstatus - the output of the **smControl status** command for all selected chassis.

Baseline:

baseline/esm.smShowDefBcGroup - the output of the **smShowDefBcGroup** command for all selected chassis.

baseline/esm.smShowSMParms - the output of the **smShowSMParms** command for all selected chassis.

During a baseline run, the above files are also created in **FF_ANALYSIS_DIR/latest**.

Full analysis:

latest/esm.smstatus - the output of the **smControl status** command for all selected chassis.

latest/esm.smShowDefBcGroup - the output of the **smShowDefBcGroup** command for all selected chassis.

latest/esm.smShowDefBcGroup.diff - **diff** of baseline and latest **smShowDefBcGroup**.

latest/esm.smShowSMParms - the output of the **smShowSMParms** command for all selected chassis latest/esm. **smShowSMParms.diff-diff** of the baseline and the latest **smShowSMParms**.

The **.diff** files are only created if differences are detected.

If the **-s** option is used and failures are detected, files related to the checks that have failed are also copied to a time stamped directory name under **FF_ANALYSIS_DIR**, such as:

```
FF_ANALYSIS_DIR/2007-11-22-09:53:04
```

5.6.6.1
Chassis SM items that are checked against the baseline

Based upon **smShowSMParms**:

- SM priority
- SM sweep rate
- SM retry and timeout
- SM fabric timeouts configured (**switchLifeTime**, **HoqLife**, **VLStall**, **PacketLifeTimes** for **PathRecords**)
- Multipath mode
  - Based on **smShowDefBcGroup**
- Default IPoIB broadcast group settings in SM (**PKey**, **MTU**, **Rate**, **SL**)

5.6.6.2
Chassis SM items also checked during healthcheck

Based upon **smControl status**:

- SM is in running state

5.6.7
all_analysis

(All):The all_analysis command has the following usage:

```
all_analysis [-b|-e] [-s] [-d dir] [-c file] [-t portsfile] [-p
ports] [-F chassisfile] [-H 'chassis'] [-G esmchassisfile] [-E
'esm chassis']
```

-b - baseline mode. The default is the compare/check mode.

-e - evaluate health only. The default is the compare/check mode.

-s - a save history of failures (i.e., errors and differences).

-d dir - a top-level directory for saving baseline and a history of failed checks. The default is **/var/opt/iba/analysis**.

-c file - an error thresholds configuration file. The default is **/etc/sysconfig/iba/iba_mon.conf**.

-t portsfile - a file with a list of local HCA ports used to access fabric(s) for analysis. The default is **/etc/sysconfig/iba/ports**.

-p ports - a list of local HCA ports used to access fabric(s) for analysis. The default is the first active port. This is specified as **hca:port**:

0:0 = 1st active port in system

0:y = port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

-F chassisfile - a file with a chassis in a cluster. The default is **/etc/sysconfig/iba/chassis**.

-H chassis - a list of chassis to execute the command on.

-G esmchassisfile - file with the SM chassis in the cluster. The default is **/etc/sysconfig/iba/esm_chassis**.

-H esmchassis - a list of SM chassis to analyze.

for example:

```
all_analysis
all_analysis -p '1:1 1:2 2:1 2:2'
```

The all_analysis command will perform the set of analysis specified in **FF_ALL_ANALYSIS**. This can be provided via the environment or via fastfabric.conf. The set of analysis which can be specified are: fabric, chassis, esm or hostsm. **FF_ALL_ANALYSIS** must be a space-separated list of the values mentioned above. These correspond to the respective analysis commands previously discussed.

Note that the **all_analysis** command has options which are a superset of the options for all other analysis commands. The options will be passed along to the respective tools (e.g., the **-c** file option will be passed on to **fabric_analysis** if it is specified in **FF_ALL_ANALYSIS**).

The output files will be all the output files for the **FF_ALL_ANALYSIS** selected set of analysis. See the previous sections for the specific output files.

Environment Variables

The following environment variables are also used by this command:

CHASSIS, CHASSIS_FILE - see the discussion on selection of chassis above.

ESM_CHASSIS, ESM_CHASSIS_FILE - see the discussion on selection of chassis above. These have the same format as **CHASSIS** and **CHASSIS_FILE**.

PORTS - list of ports, used in absence of **-t** and **-p**.

PORTS_FILE - a file containing a list of ports, used in absence of **-t** and **-p**.

FF_TIMEOUT_MULT - multiplier for response timeouts. The default is 2. This typically does not need to be set, but in the event of unexpected timeouts or extremely slow chassis or management network, a larger value can be SSH or Telnet.

FF_CHASSIS_ADMIN_PASSWORD - the password for administartor on all chassis. Used in absence of **-S** option.

FF_ANALYSIS_DIR - top level directory for baselines and failed health checks.

FF_CURTIME - time stamp to use on the directory created in **FF_ANALYSIS_DIR**. The default is the present date and time.

FF_FABRIC_HEALTH - **iba_report** options to use during a health check.

FF_CHASSIS_CMDS - a set of chassis CLI commands to get the chassis configuration.

FF_CHASSIS_HEALTH - chassis CLI command to check the chassis health.

FF_ESM_CMDS - a set of chassis CLI commands to get the SM configuration.

FF_DIFF_CMD - a Linux command to use to compare baseline to latest snapshot.

5.6.8 Manual and Automated Usage

There are two basic ways to use the tools:

- manual

- automated

In both cases the user should follow the initial setup procedure outlined above to create a good baseline of the configuration.

In the manual method, the user would run the tools manually when trying to diagnose problems, or when there is a concern or need to validate the configuration and health.

In the automated method, the user could run **`all_analysis`** or a specific tool in an automated script (such as a **`cron`** job). When run in this mode the **`-s`** option may prove useful (but care must be taken to avoid excessive saved failures). When run in automated mode, a frequency of no faster than hourly would be recommended. For many fabrics a run daily or perhaps every few hours would be sufficient. Since the exit code from each of the tools indicates the overall success/failure, an automated script could easily check the exit status and on failure email the output from the analysis tool to the appropriate administrators for further analysis and corrective action as needed.

NOTE: Running these tools too often can have negative impacts. Among the potential risks:

- Each run adds a potential burden to the SM, fabric and/or switches. For infrequent runs (hourly or daily) this impact is negligible. However, if this were to be run very frequently, the impacts to fabric and SM performance can be noticeable.
- Runs with the **`-s`** option will consume additional disk space for each run that identifies an error. The amount of disk space will vary depending on fabric size. For a larger fabric this can be on the order of 1-40 MB. Therefore, care must be taken not to run the tools too often and to visit and clean out the **`FF_ANALYSIS_DIR`** periodically. If the **`-s`** option is used during automated execution of the health check tools, it may be helpful to also schedule automated disk space checks (e.g., as a cron job).
- Runs coinciding with down time for selected components (such as servers that are offline or rebooting) will be considered failures and generate the resulting failure information. If the runs are not carefully scheduled, this could be misleading and also waste disk space.

Section 6

MPI Sample Applications

As part of a MPI Development installation, some sample MPI applications and benchmarks are installed to /opt/iba/src/mpi_apps. These can be used to perform basic tests and performance analysis of MPI and its performance.

As part of this package the following sample applications are provided:

- OSU latency (2 versions)
- OSU bandwidth (2 versions)
- OSU bidirectional bandwidth
- Pallas
- HPL
- NAS benchmarks
- tachyon demo
- mandel demo

To build the applications:

1. **`cd /opt/iba/src/mpi_apps`**
2. **`make clean`**
3. **`make full`** (builds all of the above sample applications).

Alternatives to **`full`** include:

`quick` - builds just OSU, Pallas and HPL.

`all` - builds just OSU, Pallas, HPL and NAS benchmarks.

In order to run the applications an **`mpi_hosts`** file must be created in **`/opt/iba/src/mpi_apps`** that provides the names of the hosts on which processes should be run. Depending on the value of **`VIADEV_PATH_METHOD`** in **`mpi.param`**, either IPoIB or Ethernet names can be specified, the default configuration will allow either. Typically, use of IPoIB names will provide faster job startup, especially on larger clusters.

If a host has more than one real CPU, its name may appear in the MPI hosts file once per CPU.

NOTE: Intel Xeon processors support Hyperthreading. However, for floating point intensive MPI applications, such as NAS and HPL, this feature significantly impacts performance and should be disabled.

For all but the mandel and tachyon demos, when MPI applications are run with the scripts provided, the results of the run will be logged to a file in **/opt/iba/src/mpi_apps/logs**. The file name will include the date and time of the run for uniqueness.

6.1 OSU Latency

This is a simple benchmark of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

A script is provided to run this application that will execute an assortment of sizes:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_lat**

This will run assorted latencies from 0 to 256 bytes. To run a different set of message sizes an optional argument specifying the maximum message size can be provided.

This benchmark will only use the first two nodes listed in mpi_hosts.

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.pallas config** file is used.

6.2 OSU Latency2

This is a simple performance test of end-to-end latency for various MPI message sizes. The values reported are one-direction latency.

A script is provided to run this application, which will execute an assortment of sizes:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_lat2**

This will run assorted latencies from 0-to-4Megabytes.

This benchmark will only use the first two nodes listed in **mpi_hosts**.

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.pallas config** file is used.

6.3 OSU Bandwidth

This is a simple benchmark of maximum unidirectional bandwidth.

A script is provided to run this application which will execute an assortment of sizes:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_bw**

This will run assorted bandwidths from 4K to 4Mbytes. To run a different set of message sizes an optional argument specifying the maximum message size can be provided.

This benchmark will only use the first two nodes listed in **mpi_hosts**.

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.pallas config** file is used.

6.4 OSU Bandwidth2

This is a simple benchmark of maximum unidirectional bandwidth.

A script is provided to run this application that will execute an assortment of sizes:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_bw2**

This will run assorted bandwidths from 1 byte to 4Mbytes. This benchmark will only use the first two nodes listed in **mpi_hosts**.

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.pallas config** file is used.

6.5 OSU Bidirectional Bandwidth

This is a simple benchmark of maximum bidirectional bandwidth..

A script is provided to run this application that will execute an assortment of sizes:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_bibw2**

This will run assorted bandwidths from 1 byte to 4Mbytes. This benchmark will only use the first two nodes listed in **mpi_hosts**.

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.pallas config** file is used.

6.6 High Performance Linpack (HPL)

This is a standard benchmark for Floating Point Linear Algebra performance. Included in the HPL is the Dr K. Goto Linear Algebra library. If desired, the user can modify the HPL **makefiles** to use alternate libraries. Atlas source code and the open source math library is also provided in **/opt/iba/src/mpi_apps/ATLAS**.

HPL is known to scale very well and is the benchmark of choice for identifying a systems ranking in the Top 500 supercomputers (http://www.top500.org).

Prior to running this application, a **HPL.dat** file must be installed into **/opt/iba/src/mpi_apps/hpl/bin/ICS/HPL.dat** on all nodes. The **config_hpl script** and some sample configurations are included.

The **config_hpl** script can select from one of the assorted **HPL.dat** files in **hpl-config**. For assorted cluster sizes (by number of CPUs). Assorted sample **HPL.dat** files are provided in **/opt/iba/src/mpi_apps/hpl-config**. These files are a good starting point for most clusters and should get within 10-20% of the optimal performance for the cluster. The problem sizes used assume a cluster with 1GB of physical memory per processor (e.g., for a 2 processor node, 2GB of node memory is assumed). For each cluster size, 4 files are provided:

t - a very small test run (5000 problem size)

s - a small problem size on the low end of optimal problem sizes

m - a medium problem size

l - a large problem size

These can be selected using **config_hpl**. The following command displays the preconfigured problem sizes available:

```
./config_hpl
```

For example, to quickly confirm that HPL will run on the 16 nodes in the **/opt/iba/src/mpi_apps/mpi_hosts** file, enter **./config_hpl 16t**. This will edit the **HPL.dat** file on the local host for a 16 host "very small" test, and copy that **HPL.dat** file to all hosts in the **mpi_hosts** file.

Once the **HPL.dat** has been configured and copied, HPL can be run using the script:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_hpl NP**

where **NP** is the number of processors for the run. For example:

```
./run_hpl 16
```

For more information about HPL, consult the README, TUNING and assorted HTML files in **/opt/iba/src/mpi_apps/hpl**.

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.hpl config** file is used.

6.7 Pallas

The Pallas benchmark does exhaustive benchmarking of latency and bandwidth for assorted message sizes for many MPI primitives. This benchmark is a good tool to evaluate and tune small clusters or a subset of a large cluster.

Pallas has known scalability limitations, especially in its **AllToAll** phase. This phase can simultaneously perform up to 4MB transfers to-and-from all nodes at once. The downside is a system must have approx 10*NP MB of memory available per process for Pallas data to run this benchmark. Therefore, for a small cluster (approximately 16 processors or less), it is modest at 160MB. However, for a larger cluster (approximately 256 processors or greater), it is rather large at 2.5GB.

As such, it is recommended that Pallas be used for smaller runs (2-32 processes) or that it be recognized that the benchmark is likely to fail (or swap Linux to death) at larger process counts. Depending upon the amount of memory in the system and the numbers of processes to run, the **VIADEV_MEM_REG_MAX** parameter in **/opt/iba/src/mpi_apps/mpi.param.pallas** may need to be edited.

To run pallas:

1. **cd /opt/iba/src/mpi_apps**
2. **./run_pmb NP**

where **NP** is the number of processes to run, such as:

```
./run_pmb 4
```

During this benchmark the **/opt/iba/src/mpi_apps/mpi.param.pallas config** file is used.

Appendix A
Fast Fabric Quick Install Checklist

The sections below provide a checklist to aid tracking the steps have been completed for Fabric Setup, Installation and verification. Check off each step as its performed. Refer to the section “Getting Started” on page 3-1 for a more detailed explanation of each step.

A.1 Setup The Fabric

1. Hardware installed, cabled, powered.
2. HCA installed in each server.
3. Desired Linux OS installed on each server with the following options:
   a. For root user command prompt ends in "**#** " or "**$** ".

      NOTE: There must be a space after **#** or **$**.

   b. Tcl and Expect packages installed on all IB Management Nodes.
4. Remote login as root enabled.
   a. If using SSH:
      - ensure SSH server enabled
      - all servers configured with the same root password
   b. If using RSH:
      - ensure **.rhosts** file in **/root**
      - **rsh/rlogin/rexec** are enabled
      - **/etc/securetty** file moved
5. TCP/IP Host Name Resolution
   a. If using **/etc** hosts - **/etc/hosts** file created on IB Management Node
   b. If using DNS
      - all Management Network and IPoIB hostnames added to DNS
      - **/etc/resolv.conf** file configured on IB Mangement Node.
6. NTP server setup.
7. Fabric Access Software installed on IB Management Node.

A.2

Installing and verifying Firmware on the IB Chassis

1. All Chassis connected to management network.
2. Unique IP address configured for each chassis.
3. Unique name configured for each chassis.
4. Administrator passwords configured the same on all chassis.
5. **`Fastfabric.conf`** file reviewed.
6. **`/etc/sysconfig/iba/chassis`** file created listing all chassis.
7. Chassis verified via Ethernet ping.
8. Chassis firmware updated.
9. Any additional chassis configuration completed. Such as:
   a. configuration of NTP
   b. configuration of timezone
   c. configuration of a syslog server

A.3

Installing and Configuring the Subnet Manager

1. Subnet Manager installed/enabled.
2. Subnet Manager started and running.

A.4

Installing and Verifying Firmware on the IB Switches

1. **`Fastfabric.conf`** file reviewed.
2. **`/etc/sysconfig/iba/ibnodes`** file created listing all switches and their desired names.
3. Switch firmware updated.
4. Any 9024FC switches in the IB Management Node path to fabric updated.

A.5

Install Infiniband on the Remaining Servers

1. **`Fastfabric.conf`** file reviewed.
2. **`/etc/sysconfig/iba/hosts`** file created listing all hosts except IB Management Node.
3. Verify Hosts via Ethernet ping.
4. If using RSH/RCP, verify that is RSH/RCP configured.

5. Setup password-less SSH/SCP.
6. TCP/IP Name Resolution configured on all hosts.
   a. if using **`/etc/hosts`** - Copy **`/etc/hosts`** to all hosts.
   b. if using DNS - **`/etc/resolv.conf`** copied or configured on all hosts.
7. Show **`uname -a`** for all hosts and verify OS version installed.
8. Install QuickSilver software.
9. Configure IPoIB IP Addresses.
10. Build MPI Test Apps and copy to hosts.
11. Reboot hosts.
12. Any additional host configuration completed.

A.6 Verifying Infiniband on the Remaining Servers

1. **`Fastfabric.conf`** file reviewed.
2. **`/etc/sysconfig/iba/allhosts`** file created listing all hosts including IB management nodes.
3. Verify hosts via Ethernet ping.
4. Summary of fabric components reviewed.
5. Port state and symbol errors checked via **`iba_report`** to show status of host IB ports.
6. Verify hosts can see each other.
7. Verify hosts ping via IPoIB.
8. Refresh SSH known hosts.
9. Check MPI performance.

A.7 Complete Installation of additional IB Management Nodes

For Each IB management node:

1. Install additional Fabric Access software components.
2. Copy Fast Fabric configuration files from initial IB management node and edit host lists.
3. Setup password-less SSH/SCP.
4. Refresh SSH known hosts.

A.8

Configure and initialize health check tools

This procedure should be followed on each IB management node from which the health check tools will be used.

1. Edit **`fastfabric.conf`** and review the health check tools parameters.
2. If using embedded SM(s), create **`/etc/sysconfig/iba/esm_chassis`**.
3. Perform a health check using **`all_analysis -e`**.
4. Create a cluster configuration baseline using **`all_analysis -b`**.
5. If desired, schedule regular runs of **`all_analysis`** via **`cron`** or other mechanisms.

Appendix B
Fast Fabric Configuration Files

The following configuration files are used by Fast Fabric:.

Table B-1. Fast Fabric Configuration Files

Configuration File	Description
/etc/sysconfig/fastfabric.conf	Overall configuration file
/etc/sysconfig/iba/iba_mon.conf	Error thresholds
/etc/sysconfig/iba/allhosts	List of all hosts managed by fast fabric including the localhost
/etc/sysconfig/iba/hosts	List of all hosts managed by fast fabric except the localhost
/etc/sysconfig/iba/chassis	List of all chassis managaed by fast fabric
/etc/sysconfig/iba/ibnodes	List of all externally managed switches managed by fast fabric
/etc/sysconfig/iba/esm_chassis	List of all chassis running an embedded SM which are to be monitored via esm_analysis
/etc/sysconfig/iba/ports	List of local HCA ports (eg. subnets) to be used for fabric health analysis

B.1 fastfabric.conf

The Fast Fabric tools support a configuration file **`/etc/sysconfig/fastfabric.conf`**. This file can be used to provide default settings for most of the Fast Fabric command line options. The config file is a bash shell script which will be included by each tool. As such the file should be implemented such that environment variables defined before the config file is executed will not be altered. The sample displayed below makes use of the bash syntax:

```
var= "${var:-value}"
```

such that only uninitialized variables are overwritten by the configuration file.

A sample file is provided below, and matches the internal defaults of the Fast Fabric tools. For a given release consult **`/etc/sysconfig/fastfabric.conf-sample`** for a sample file with the defaults of the given release. If **`fastfabric.conf`** does not assign a value to a given configuration variable, the default value will be used.

NOTE: Do not edit **`/etc/sysconfig/fastfabric.conf-sample`**.

The use of various configuration variables are discussed in the Environment Variables section for each command.

```
#!/bin/bash
# [ICS VERSION STRING: unknown]
# This is a bash sourced config file which defines variables used
in
# fast fabric tools.  Command line arguments will override these
settings.
# Assignments should be scripted such that this file does not
override
# exported environment settings, as shown in the defaults below

if [ "$CONFIG_DIR" = "" ]
then
        if [ -d /etc/sysconfig ]
        then
                CONFIG_DIR=/etc/sysconfig
        else
                CONFIG_DIR=/etc
        fi
        export CONFIG_DIR
fi

# Override default location for HOSTS_FILE
export HOSTS_FILE="${HOSTS_FILE:-$CONFIG_DIR/iba/hosts}"

# Override default location for CHASSIS_FILE
export CHASSIS_FILE="${CHASSIS_FILE:-$CONFIG_DIR/iba/chassis}"

# Override default location for ESM_CHASSIS_FILE
export
ESM_CHASSIS_FILE="${ESM_CHASSIS_FILE:-$CONFIG_DIR/iba/esm_chassis}
"

# Override default location for IBNODES_FILE
export IBNODES_FILE="${IBNODES_FILE:-$CONFIG_DIR/iba/ibnodes}"

# Override default location for PORTS_FILE
export PORTS_FILE="${PORTS_FILE:-$CONFIG_DIR/iba/ports}"

# Default suffix for IPoIB host names
# the special value "NONE" is used to indicate a value of ""
export FF_IPOIB_SUFFIX="${FF_IPOIB_SUFFIX:--ib}"

# Default management host
export MGMT_HOST="${MGMT_HOST:-localhost}"
```

```
# The shell functions below are only defined if no existing
function/command
# with given name, hence allowing use of shell functions or
creation of a
# command for this operation

# shell Function to convert a basic hostname into an IPoIB
hostname
# if FF_IPOIB_SUFFIX is "", this should return $1 unmodified
# such that commands can be used with -i "" to skip IPoIB
operations
# (special value of NONE will be converted to "" before this is
called)
if ! type ff_host_basename_to_ipoib >/dev/null 2>/dev/null
then
        function ff_host_basename_to_ipoib()
        {
                # $1 = hostname provided
                echo "$1$FF_IPOIB_SUFFIX"
        }
fi

# shell Function to convert a hostname into a basic hostname
# (eg. remove IPoIB suffix, etc) should match result of "hostname
-s" on host
# if FF_IPOIB_SUFFIX is "", this should return $1 unmodified
# such that commands can be used with -i "" to skip IPoIB
operations
# (special value of NONE will be converted to "" before this is
called)
if ! type ff_host_basename >/dev/null 2>/dev/null
then
        function ff_host_basename()
        {
                # $1 = hostname provided (could be ethernet or
IPoIB name)
                echo "$1"|sed -e "s/$FF_IPOIB_SUFFIX\$//"
        }
fi

# IP netmask for IPoIB subnet [-m option]
# if "" default will be determined based on class of IP address
[A, B, C]
export FF_IPOIB_NETMASK="${FF_IPOIB_NETMASK:-}"

# Maximum parallel processes for ibtest and -p option on other
commands
```

```
# set to 1 to avoid parallel execution
export FF_MAX_PARALLEL=${FF_MAX_PARALLEL:-20}

# If the systems are slow for some reason, this can be used to
provide a
# multiplier for all timeouts in ibtest
export FF_TIMEOUT_MULT=${FF_TIMEOUT_MULT:-2}

# InfiniServ product to install during ibtest load and ibtest
upgrade
export FF_PRODUCT="${FF_PRODUCT:-InfiniServPerf}"

# InfiniServ packages to install during ibtest load [-P option]
export FF_PACKAGES="${FF_PACKAGES:-iba ipoib mpi}"

# Additional INSTALL options during ibtest load [-I option]
export FF_INSTALL_OPTIONS="${FF_INSTALL_OPTIONS:-}"

# Additional INSTALL options during ibtest upgrade [-U option]
export FF_UPGRADE_OPTIONS="${FF_UPGRADE_OPTIONS:-}"

# where to upload server specific files to during uploadall,
captureall
# [-d option]
export UPLOADS_DIR="${UPLOADS_DIR:-./uploads}"

# where to download server specific files from during downloadall
# [-d option]
export DOWNLOADS_DIR="${DOWNLOADS_DIR:-./downloads}"

# where to save history and baselines for cluster analysis and
health checks
# [-d option]
export FF_ANALYSIS_DIR="${FF_ANALYSIS_DIR:-/var/opt/iba/analysis}"

# How to login to host systems during ibtest
# can be ssh, rsh or telnet
# Note: at present cmdall and scpall always assume password-less
ssh/scp
# Note: check_rsh always assumes password-less rsh/rcp
# Note: arguments (-s) to setup_ssh select ssh setup via rsh/rcp
or ssh/scp
export FF_LOGIN_METHOD="${FF_LOGIN_METHOD:-ssh}"

# initial user name (cannot be root if CFG_LOGIN_METHOD=telnet)
export FF_USERNAME="${FF_USERNAME:-root}"

# password for FF_USERNAME (not needed if password-less ssh/rsh
configured)
```

```
export FF_PASSWORD="${FF_PASSWORD:-}"

# if FF_USERNAME is not root, what is the root password needed
when
# suing to root
export FF_ROOTPASS="${FF_ROOTPASS:=}"

# How to login to chassis
# can be ssh or telnet
export
FF_CHASSIS_LOGIN_METHOD="${FF_CHASSIS_LOGIN_METHOD:-telnet}"

# password for admin (not needed if password-less ssh configured)
export
FF_CHASSIS_ADMIN_PASSWORD="${FF_CHASSIS_ADMIN_PASSWORD:-adminpass}
"

#base device number for IPoIB devices - IPoIB devices in the
systems will
# be named "ib($FF_IPOIB_BASE_DEV_NUM)
ib($FF_IPOIB_BASE_DEV_NUM+1) ..."
# On Silvertstorm stack it will be "ib1 ib2 ...."
# On OFED stack it will be "ib0 ib1 ..."
export FF_IPOIB_BASE_DEV_NUM="${FF_IPOIB_BASE_DEV_NUM:-1}"

# shell Function to return the base IPoIB device number for this
stack type.
# For Silverstorm stack installations this function should return
1, for
# OFED stack installations this should return 0.
if ! type ff_ipoib_base_dev_num >/dev/null 2>/dev/null
then
        function ff_ipoib_base_dev_num()
        {
                # No arguments to this function
        # $host is supplied as an arguemnt when this function is
called
        # but it is ignored here. This is a hack to limit the
number of
                # changes required.
                echo "$FF_IPOIB_BASE_DEV_NUM"
        }
fi

# options for iba_report during fabric_analysis health check
# alternative would be to use "-i seconds" instead of -C
# depending on fabric design, more strict link speed checks, such
as
# -o slowconfiglinks or -o slowconnlinks may be appropriate
```

```
export FF_FABRIC_HEALTH="${FF_FABRIC_HEALTH:- -s -C -o errors -o
slowlinks}"

# list of CLI commands to issue during chassis_analysis
export FF_CHASSIS_CMDS="${FF_CHASSIS_CMDS:-showInventory fwVersion
showIBNodeDesc ismShowPStatThresh ismChassisSet12x timeZoneConf
timeDSTConf snmpCommunityConf snmpTargetAddr showChassisIpAddr
showDefaultRoute}"
# other possible additions (if running newer chassis FW which
supports these)
# ismIslSet12x, ismIslSetSpeed

# single CLI command to issue to check overall health during
chassis_analysis
# hwCheck is prefered, but is not supported on older chassis FW
export FF_CHASSIS_HEALTH="${FF_CHASSIS_HEALTH:-hwCheck}"
#export FF_CHASSIS_HEALTH="${FF_CHASSIS_HEALTH:-fruInfo}"

# list of CLI commands to issue during esm_analysis
export FF_ESM_CMDS="${FF_ESM_CMDS:-smShowSMParms
smShowDefBcGroup}"

# list of analysis to perform during all_analysis
# pick appropriate type of SM to analyze
#export FF_ALL_ANALYSIS="${FF_ALL_ANALYSIS:-fabric chassis hostsm
esm}"
export FF_ALL_ANALYSIS="${FF_ALL_ANALYSIS:-fabric chassis}"

# command to perform text files for baseline vs latest snapshot
during
# fabric health analysis commands
#export FF_DIFF_CMD="${FF_DIFF_CMD:-diff}"
export FF_DIFF_CMD="${FF_DIFF_CMD:-diff -C 1}"
```

B.2 iba_mon.conf

The **/etc/sysconfig/iba/iba_mon.conf** configuration file defines port statistics thresholds for use by **iba_report**, **fabric_analysis**, **all_analysis** and **iba_mon**.

This file lists a threshold for each port statistic. If the threshold for a given statistic is not defined or is set to **0**, the given statistic will not be checked.

A sample file is provided below, and matches the internal defaults of the Fast Fabric tools. For a given release consult **/etc/sysconfig/iba/iba_mon.conf-sample** for a sample file with the defaults of the given release.

NOTE: Do not edit **`/etc/sysconfig/iba/iba_mon.conf-sample`**.

```
# This file controls the iba_mon Port Counter monitoring
Thresholds.
# [ICS VERSION STRING: unknown]
#
# Error Counters are specified in absolute number of errors over
Interval.
# All Data Movement thresholds are specified in terms of average
data/second
# over the monitoring interval.
#
# Setting a threshold to 0 disables monitoring of the given
counter
#
# Output is generated when a threshold is exceeded.
#
# Counters for which a non-zero threshold is specified will be
cleared by
# iba_mon and may impact any remote Performance Managers which are
monitoring
# the given Counter

Interval                          10      # monitoring interval in
seconds

SyslogFacility              local6      # syslog facility code, or
disable

# Normal Data Movement
PortXmitData                       0      # as MB/second
PortRcvData                        0      # as MB/second
PortXmitPkts                       0      # as packets/second
PortRcvPkts                        0      # as packets/second

# Error Counters
SymbolErrorCounter               100
LinkErrorRecoveryCounter           3
LinkDownedCounter                  3
PortRcvErrors                    100
PortRcvRemotePhysicalErrors      100
#PortRcvSwitchRelayErrors        100 # Anafa2 issue, incorrectly
increments
PortXmitDiscards                 100
PortXmitConstraintErrors          10
PortRcvConstraintErrors           10
LocalLinkIntegrityErrors           3
ExcessiveBufferOverrunErrors       3
VL15Dropped                      100
```

B.3 Host List Files

The **/etc/sysconfig/iba/hosts** and **/etc/sysconfig/iba/allhosts** files are used to specify the hosts which Fast Fabric will operate against for many operations.

If desired alternate filenames may be specified in **fastfabric.conf**, via environment variables or on the command line. Refer to the section "Selection of Hosts" on page 5-3 for more information.

Below is a sample host list file:

```
# this is a comment
192.168.0.4# host identified by IP address
n001# host identified by resolvable TCP/IP name
include /etc/sysconfig/iba/hosts-mpi# included file
```

Each line of the host list file may specify a single host, a comment or another host list file to include.

Hosts may be specified by IP address or a resolvable TCP/IP hostname. Typically hostnames are used for readability. Also, some Fast Fabric tools will translate the supplied host names to IPoIB hostnames, in which case names are generally easier to translate than numeric IP addresses. Typically management network host names are specified. However, if desired, IPoIB hostnames or IP addresses may be used. This can accelerate large file transfers and other operations.

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute path names. If relative path names are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line by using a **#** to precede the comment. On lines with hosts or include directives, the **#** must be white-space separated from any preceding host name, IP address or included file name.

B.4 Chassis List Files

The **/etc/sysconfig/iba/chassis** and **/etc/sysconfig/iba/esm_chassis** files are used to specify the SilverStorm IB chassis that Fast Fabric will operate against for many operations.

If desired alternate filenames may be specified in **fastfabric.conf**, via environment variables or on the command line. Refer to the sectoin "Selection of Chassis" on page 5-4 for more information.

Below is a sample chassis file:

```
# this is a comment
192.168.0.5# chassis IP address
edge1# chassis resolvable TCP/IP name
include /etc/sysconfig/iba/corechassis# included file
```

Each line of the chassis list file may specify a single chassis, a comment or another chassis list file to include.

Chassis may be specified by chassis management network IP address or a resolvable TCP/IP name. Typically, names are used for readability.

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute path names. If relative path names are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line by using a **#** to precede the comment. On lines with chassis or include directives, the **#** must be white-space separated from any preceding name, IP address or included filename.

B.5 Selection of slots within a chassis

Normally, operations are performed against the management card in the chassis. For operations such as **cmdall**, the command is executed against the management interface for the given chassis. For more sophisticated operations, such as firmware update, a directory with firmware for each chassis card type can be supplied and all cards in the chassis will be updated with the appropriate firmware from that directory.

However, in some cases it may be desirable to perform operations against a specific subset of cards within the chassis. In this case the chassis IP address, name within a chassis list or a chassis file can be augmented with a list of slot numbers to operate on. This is done in the form:

```
chassis:slot1,slot2,…
```

For example:

```
i9k229:0
i9k229:0,1,5
192.168.0.5:0,1,5
```

NOTE: There must be no spaces within the chassis name and/or slot list.

This format is used by **cmdall** and **chassis firmware update**. This format may be used anywhere a chassis name or IP address is valid, such as the **-H** option, the **CHASSIS** environment variable or **chassis list files**. The slot number specified is ignored on some operations (such as **pingall**). Only slots containing management cards, EVICs and FVICs may be specified with this format. For all

9000 series chassis, slot **0** is always an alias for the presently active management card for the chassis. For the remainder of slot usages in the chassis, the **chassisQuery** command can be executed against a given chassis to identify which slots have management, EVIC or FVIC cards.

NOTE: For any operation, care should be taken that a given chassis is listed only once with all relevant slots as part of that single specification. This is important so that parallel operations do not cause conflicting concurrent operations against a given chassis.

B.6 Switch List Files

The **/etc/sysconfig/iba/ibnodes** file is used to specify the externally-managed SilverStorm IB switches that Fast Fabric will operate against for many operations.

If desired alternate file names may be specified in **fastfabric.conf**, via environment variables or on the command line. Refer to the section "Selection of Switches" on page 5-7 for more information.

Below is a sample switch list file:

```
# this is a comment
0x00066a00d9000138,i9k138# Node GUID with desired Name
0x00066a00d9000139,i9k139# Node GUID with desired Name
include /etc/sysconfig/iba/moreswitches# included file
```

Each line of the switch list file may specify a single switch, a comment or another switch list file to include.

Switches can be specified by node GUID optionally followed by a coma and the IB Node Description (i.e., the name) to be assigned to the switch. The GUID will be used to select the switch and on firmware update operations, the node description will be written to the switch such that other Fast fabric tools (such as **saquery** and **iba_report**) can provide a more easily readable name for the switch.

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute path names. If relative path names are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line. By using a **#** to precede the comment. On lines with chassis or include directives, the **#** must be white-space separated from any preceding GUID, name or included file name.

It is recommended that a unique node description be specified for each switch. This name should follow typical naming rules and use the characters a-z, A-Z, 0-9, and underscore. No spaces are allowed in the node description. Additionally, names should not start with a digit.

For externally-managed switches, the node GUID can be found on a label on the bottom of the switch. Alternately the node GUIDs for switches in the fabric can be found use a command such as:

```
saquery -t sw -o nodeguid
```

NOTE: The above command will report all switch node GUIDs, including those of internally-managed chassis such as the SilverStorm 9120. GUIDs for internally-managed chassis cannot be specified for use in the **ibnodes** file.

B.7 Port List Files

The **/etc/sysconfig/iba/ports** file is used to specify the local HCA ports (i.e., subnets) that Fast Fabric will use for fabric health analysis.

If desired, alternate filenames may be specified in **fastfabric.conf**, via environment variables or on the command line. Refer to the section "Selection of local Ports (subnets)" on page 5-9 for more information.

Below is a sample port list file:

```
# this is a comment
1:1# first port on 1st HCA
1:2# second port on 1st HCA
2:1# first port on 2nd HCA
3:0# first active port on 3rd HCA
include /etc/sysconfig/iba/ports-plain2# included file
```

Each line of the port list file may specify a single port, a comment or another port list file to include.

Ports are specified as **hca:port**. No spaces are permitted. The first HCA is **1** and the first port is **1**. The special value **0** for HCA or port has special meaning. The allowed formats are:

0:0 = 1st active port in system

0:y = port y within system

x:0 = 1st active port on HCA x

x:y = HCA x, port y

Files to be included may be specified via an **include** directive followed by a file name. File names specified should generally be absolute path names. If relative path names are used, they will be searched for within the current directory then **/etc/sysconfig/iba**.

Comments may be placed on any line. By using a **#** to precede the comment. On lines with a port or include directive, the **#** must be white-space separated from any preceding port or included file name.

Appendix C
Configuration of IPoIB Name Mapping

The Fast Fabric tools support the concept of a management network and an IPoIB network. For some clusters the management network will be a low speed network such as 10/100 Ethernet. For other clusters IPoIB may serve double duty as the host management network.

NOTE: When using IPoIB as the management network, the initial installation of IB software cannot be done via Fast Fabric.

The various Fast Fabric tools will translate from host names provided to and from IPoIB names as needed. This permits the host names given to be either management network or IPoIB network names. The default configuration file assumes that IPoIB host names are formed by adding a **`-ib`** suffix to the management network name. If a different suffix is desired, **`FF_IPOIB_SUFFIX`** can be changed. If IPoIB is also being used as the management network, **`FF_IPOIB_SUFFIX`** can be set to an empty string `""`.

The translation is driven by the following functions within **`fastfabric.conf`**:

```
ff_host_basename - given a management network or IPoIB hostname,
translate to management network name, should match hostname -s
ff_host_basename_to_ipoib - given a management network name,
translate to IPoIB hostname
```

More complex mappings can be specified by implementing alternate algorithms for these functions.

NOTE: When managing a cluster where compute nodes are not running the QuickSilver host stack or where the IPoIB settings on the compute nodes are incompatible with the IB Management node (for example when a 4K MTU is used on the compute nodes), it is recommended not to run IPoIB on the IB management nodes.

Appendix D
Multi-Subnet Fabrics

Fast Fabric is designed primarily to manage a single subnet fabric. However many powerful functions of FastFabric are also available when installing and operating multi-subnet fabrics.

When operating a multi-subnet fabric, an subnet manager (SM) is required for each subnet. An SM may be run within switches within each subnet, or a host-based SM may be run. A host-based SM can manage multiple subnets (assuming the host server is connected to more than one subnet).

For multi-subnet fabrics a number of combinations are possible:

1. **All subnets are completely independent (except for any interconnecting routers)**: If a separate Fast Fabric node is being used per subnet and servers are not installed in more than one subnet, the individual subnets can be treated completely separately. In this case, follow all the previous Fast Fabric instructions for each fabric.
2. **The subnets are primarily independent**: If the only IB components common to more than one subnet are the Fast Fabric node (and possibly SM nodes) and no servers are installed in more than one subnet, consult the instructions below for *Primarily Independent Subnets*.
3. **The subnets are overlapping**: If multiple IB components are common to more than one subnet, such as Fast Fabric node(s), servers, etc., consult the instructions below for *Overlapping Subnets*.

D.1 Primarily Independent Subnets

If the Fast Fabric node (and possible SM nodes) is the only common IB servers between subnets, Fast Fabric may be used to assist in server installation and fabric operation. Follow the installation instructions outlined in “Getting Started” on page 3-1 with the following adjustments:

From “Design the Fabric” on page 3-1, design the cabling such that the Fast Fabric node will be connected to each IB subnet it will manage. The Fast Fabric node must also have a management network path to all the nodes in all the subnets that it will manage. As part of the design consider where routes between subnets are desired between IB routers, IPoIB routers, etc.

“Design the Fabric” on page 3-1 can be performed as per the instuctions. When installing the Fabric Access Software on the IB management node, IPoIB will need to be configured such that each IB subnet is an independent IPoIB network interface,

typically with different IP subnets. Consult the *QuickSilver Fabric Access Software Users Guide* for more information on configuring IPoIB.

NOTE: When managing a cluster where compute nodes are not running the QuickSilver host stack or where the IPoIB settings on the compute nodes are incompatible with the IB Management node (e.g., when a 4K MTU is used on the compute nodes), it is recommended not to run IPoIB on the IB management node(s).

"Installing and Verifying Firmware on the SilverStorm IB Chassis" on page 3-6 can be performed as per the instructions. When creating the chassis file, list all SilverStorm 9000 series internally-managed IB switches in all subnets. If desired, additional files may also be created per subnet that list only the SilverStorm chassis in each subnet.

"Installing and Configuring the Subnet Manager" on page 3-9 can be performed as per the instructions. At least one subnet manager is required per IB subnet. Consult the *QuickSilver Fabric Manager and Fabric Viewer Users Guide* for more information on how to configure a host SM node to manage more than one IB subnet.

"Installing and Verifying Firmware on the IB Switches" on page 3-10. At this time this operation is not supported for IB management nodes connected to more than one IB subnet.

"Installing InfiniBand on the Remaining Servers" on page 3-12 can be performed as per the instructions. When creating the **`hosts`** file, list all of the hosts in all subnets except the IB management node where Fast Fabric is being run. If desired, additional files may also be created per subnet that list the hosts in each subnet (except the IB management node).

"Verifying InfiniBand on the Remaining Servers" on page 3-16 has the following adjustments from the instructons.

- **(All)**: Create the **`allhosts`** file per the instructions. Next, create additional files per subnet that list all the hosts in each subnet including the IB management node.
- **(All)**: "Verify Hosts via Ethernet ping" on page 4-4 can be performed per the instructions.
- **(All)**: "Summary of Fabric Components" on page 4-9 is not presently supported for IB management nodes connected to multiple fabrics. Instead manually run:

  ```
  iba_report -d 0 -h x -p y
  ```

  where **`x`** and **`y`** specify the applicable HCA and port to select the desired subnet. Repeat for each subnet.
- **(Host)**: "Show Status of Host IB Ports" on page 4-9 may be run per the instructions.

- **(All)**: However instead it is recommended to run:

  ```
  iba_report -i 10 -o errors -o slowlinks -h x -p y
  ```

  where **x** and **y** specify the applicable HCA and port to select the desired subnet. Repeat for each subnet.

- **(Host)**: "Verify Hosts see each other" on page 4-9 can be run for each subnet by using the **allhosts** files specific to each subnet (i.e., those listing only hosts in a single subnet).
- **(Host)**: "Verify Hosts ping via IPoIB" on page 4-9 and **(Linux)**: "Refresh SSH Known Hosts" on page 4-9 may be run per the instructions.
- **(Host)**: "Check MPI Performance" on page 4-10 can be run for each subnet by using the **allhosts** files specific to each subnet (i.e., those listing only the hosts in a single subnet).

"Complete Installation of additional IB Management Nodes" on page 3-18 can be performed as per the instructions. When copying Fast Fabric configuration files to the additional IB management nodes, be sure to also copy the additional hosts, chassis and **allhosts** files that were created per subnet.

NOTE: In assymetrical configurations where the IB management nodes are not all connected to the same set of subnets, the files copied to each management node may need to be slightly different. For example configuration files for **fabric_analysis** may indicate different port numbers or host files used for Fast Fabric and MPI may need to list different hosts.

"Configure and Initialize Health Check Tools" on page 3-19 can be performed as per the instructions. Additionally, create a **/etc/sysconfig/iba/ports** file listing each of the IB management node local HCAs and ports that are connected to a unique subnet. When running fabric_analysis or all_analysis, the default will be to use the **ports** file. If desired, the **-p** and **-t** options or the **PORTS/PORTS_FILE** environment variables may be used to specify all the HCAs and ports on the IB management node such that all subnets are checked. Similarly, the **esm_chasssis** and chassis files used should list all relevant SilverStorm IB chassis in all subnets.

"Running HPL" on page 3-20 can be run for each subnet by creating **mpi_hosts** files specific to each subnet (i.e., only listing hosts in a single subnet).

"Upgrading IB software" on page 3-21 can be performed as per the instructions.

D.2 Overlapping Subnets

If multiple IB components are common between subnets (in addition to the IB management nodes), Fast Fabric may be used to assist in server installation and

fabric operation. Follow the installation instructions outlined in “Getting Started” on page 3-1 with the following adjustments:

From “Design the Fabric” on page 3-1, design the cabling such that the Fast Fabric node will be connected to each IB subnet it will manage. The Fast Fabric node must also have a management network path to all the nodes in all the subnets it will manage. As part of the design consider where routes between subnets are desired, between IB routers, IPoIB routers, etc.

“Set Up the Fabric” on page 3-2 can be performed as per the instuctions. When installing the Fabric Access software on the IB Management node, IPoIB will need to be configured such that each IB subnet is an independent IPoIB network interface, typically with different IP subnets. Consult the *QuickSilver Fabric Access Software Users Guide* for more information on configuring IPoIB.

NOTE: When managing a cluster where compute nodes are not running the QuickSilver host stack or where the IPoIB settings on the compute nodes are incompatible with the IB management node (e.g., when a 4K MTU is used on the compute nodes), it is recommended not to run IPoIB on the IB management node(s).

“Installing and Verifying Firmware on the SilverStorm IB Chassis” on page 3-6 can be performed as per the instructions. When creating the chassis file, list all SilverStorm 9000 series internally-managed IB switches in all subnets. If desired, additional files may also be created per subnet that list only the SilverStorm chassis in each subnet.

“Installing and Configuring the Subnet Manager” on page 3-9 can be performed as per the instructions. At least one subnet manager is required per IB subnet. Consult the *QuickSilver Fabric Manager and Fabric Viewer Users Guide* for more information on how to configure a host-based SM node to manage more than one IB subnet.

“Installing and Verifying Firmware on the IB Switches” on page 3-10. At this time this operation is not supported for IB management nodes connected to more than one IB subnet.

“Installing InfiniBand on the Remaining Servers” on page 3-12 can be performed as per the instructions. When creating the hosts file, list all the hosts in all subnets except the IB management node where Fast Fabric is being run. If desired, additional files may also be created per subnet that list the hosts in each subnet (except the IB management node).

For hosts that are connected to more than one IB subnet, IPoIB will need to be configured such that each IB subnet is an independent IPoIB network interface, typically with different IP subnets. Consult the *QuickSilver Fabric Access Software Users Guide* for more information on configuring IPoIB.

“Verifying InfiniBand on the Remaining Servers” on page 3-16 has the following adjustments from the instructons.

- **(All)**: Create the **allhosts** file per the instructions. Next, create additional files per subnet that list all the hosts in each subnet including the IB management node.
- **(All)**: "Verify Hosts via Ethernet ping" on page 4-4 can be performed per the instructions.
- **(All)**: "Summary of Fabric Components" on page 4-9 is not presently supported for IB management nodes connected to multiple fabrics. Instead manually run:

  ```
  iba_report -d 0 -h x -p y
  ```

  where **x** and **y** specify the applicable HCA and port to select the desired subnet. Repeat for each subnet.
- **(Host)**: "Show Status of Host IB Ports" on page 4-9 may be run per the instructions.
  - **(All)**: However, instead it is recommended to run:

  ```
  iba_report -i 10 -o errors -o slowlinks -h x -p y
  ```

    where **x** and **y** specify the applicable HCA and port to select the desired subnet. Repeat for each subnet.
- **(Host)**: "Verify Hosts see each other" on page 4-9 can be run for each subnet by using the **allhosts** files specific to each subnet (i.e., those only listing hosts in a single subnet).
- **(Host)**: "Verify Hosts ping via IPoIB" on page 4-9 and **(Linux)**: "Refresh SSH Known Hosts" on page 4-9 may be run per the instructions.
- **(Host)**: "Check MPI Performance" on page 4-10 can be run for each subnet by using the **allhosts** files specific to each subnet (i.e., those listing only the hosts in a single subnet).

"Complete Installation of additional IB Management Nodes" on page 3-18 can be performed as per the instructions. When copying Fast Fabric configuration files to the additional IB management nodes, be sure to also copy the additional hosts, chassis and **allhosts** files created per subnet.

NOTE: In assymetrical configurations where the IB management nodes are not all connected to the same set of subnets, the files copied to each management node may need to be slightly different. For example, configuration files for **fabric_analysis** indicating different port numbers or host files used for Fast Fabric and MPI may need to list different hosts.

"Configure and Initialize Health Check Tools" on page 3-19 can be performed per the instructions. In addition, create a **/etc/sysconfig/iba/ports** file listing the IB management node local HCAs and ports that are connected to a unique subnet. When running **fabric_analysis** or **all_analysis**, the default is to use the ports file. If desired, the **-p** and **-t** options or the **PORTS/PORTS_FILE** environment

variable may be used to specify all the HCAs and ports on the IB management node such that all subnets are checked. Similarly, the **esm_chasssis** and chassis files used should list all relevant SilverStorm IB chassis in all subnets.

"Running HPL" on page 3-20 can be run for each subnet by creating **mpi_hosts** files specific to each subnet (i.e., only listing hosts in a single subnet).

"Upgrading IB software" on page 3-21 can be performed per the instructions.